COMPUTERZIED BRAIN TUMOR BOUNDARY DETECTION : A GENETIC ALGORITHM BASED APPROACH

by

ARINDAM BANERJEE

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

April 1999

COMPUTERZIED BRAIN TUMOR BOUNDARY DETECTION: A GENETIC ALGORITHM BASED APPROACH

A Thesis Submitted

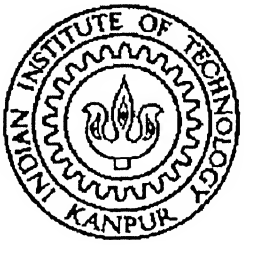
in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

ARINDAM BANERJEE



to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

April, 1999

Certificate

It is certified that the work contained in the thesis entitled "COMPUTERIZED BRAIN TUMOR BOUNDARY DETECTION: A GENETIC ALGORITHM BASED APPROACH", by Arindam Banerjee, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

26.4.99

(Dr. G. C. Ray)
Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

April, 1999.

Acknowledgements

I would like to express my sincere gratitude to Dr. G. C. Ray for his constant encouragement in completing my thesis work. I am grateful to him for the amount of support and freedom he gave me while carrying out the thesis. I shall fondly remember the time spent with him and his family.

I want to extend my special thanks to Sandy who was always beside me. Among other things , he introduced me to the linux revolution. The setting up, maintenance and time-to-time upgradation of our linux server *Sadhana* was a very linuxically enlightening and satisfying experience. I also want to extend my special thanks to Siddhartha, who was my partner in my surrealistic adventures. We shared a nice time together over the the paintings of Dali, Eshar and Picasso, the songs of Tagore, 'Maya', Gulzar, Dylan and Denver, the music of Chaurasia, Bismillah and Bach, and the waters of our swimming pool.

I would love to mention the names of my fellow IEEE (Institute of Excessive Eating Engineers) members Siddhartha, Deepanjan, Mama, Sandy and Kousik. I will always remember the delicious evenings spent with them at Baba and Danapani. I would also like to extend my thanks to Karthik, the born entertainer and analogy master, and Ashokji, the soft-spoken gentleman.

I would take this opportunity to mention the names of two great persons I have known in my stay at IITK. One is Shafi – the answer to all computer-related problems. The other is Sir Bnaaru, the gorilla, who is sure to become a Noble laureate one day. I would also like to thank Ritesh, Jay, Pratul, Mashuq, Vinayji, Ganguly, Barman and others who have made my stay at IITK a Joyful experience.

Abstract

Detection of exact brain tumor boundaries from computerized brain images is a basic problem in medical imaging. The boundary detection problem has been formulated as an optimization process that seeks the boundary points in order to minimize an energy function based on an active contour model.

A modified version of the standard genetic algorithm forms the basis for solving the optimization problem. Morphological preprocessing leads to the formation of a population of approximate boundaries of the tumor. The main algorithm deals with a population of boundary contours, performing genetic operations on them. *Selection* and formation of mating pool is done using stochastic remainder selection for less noisy results followed by a roulette wheel selection. *Spatial crossover* is implemented in a non-standard way to give rise to a single child contour from two parent contours, the process being repeated for all randomly selected pairs of virgin parents from the mating pool. *Mutation* is done in a deterministic manner on each contour thereby pushing them to the nearest local minima in the energy landscape. This process is repeated for some generations, and since the algorithm leads to the *survival of the fittest* under a given condition, the condition in this case being that of having minimum energy, the exact boundary of the tumor emerges after some generations.

The method is quite insensitive to noise because of the morphological preprocessing. As an optimizer, the algorithm is extremely robust since it deals with a population of possible solutions. The effectiveness of the approach has been shown by experiments on different slices of a magnetic resonance imaging data set.

Contents

List of Figures

Chapter 1

Introduction

An important step in most medical imaging analysis systems is to extract the boundary of an area we are interested in. Possible applications include the delineation of brain tumors in computed tomography (CT) or magnetic resonance imaging (MRI) sequences for planning radiation therapy treatment, the extraction of epicardial and endocardial boundaries of the left ventricle for studying cardiac functions such as the pressure-volume ratio, and the volumetric measurement of absolute white and gray matters for the study of degenerative diseases.

In MRI, one of the principal regions of interest is the brain. Currently in many clinical applications, the boundary of a tumor in a head image is usually traced by hand. This manual approach is prone to errors and becomes infeasible when dealing with large data sets. Therefore, there is a great need for a computerized system to perform the tumor boundary detection task. An automated computerized system can be coupled with delicate surgical tools, including radiosurgery, to develop accurate and reliable techniques for brain tumor detection and removal. In the following section we take a brief look at the recent trends in neurosurgery in applications related to brain

tumor detection and removal.

1.1 Recent trends in Neurosurgery

1.1.1 Surgery

Current trends in neurosurgery focus around imaged-guided surgery — ways of combining three-dimensional computer-assisted technology and surgical tools. Collectively, these tools allow neurosurgeons the ability to plan and execute a surgical procedure in three dimensional space.

Many tools take advantage of this technology. *Holography* helps the neurosurgeon plan and practice a surgical approach prior to the actual procedure by giving three-dimensional depth to scans. *Real-time MRI* units allow the neurosurgeon to use MRI guidance during surgery to verify the amount of tumor removal. *Stereotactic navigational systems* make use of a hand-held wand (the ISG Viewing Wand). When touched to the brain, the wand transmits three-dimensional coordinates to a computer screen, showing the location of the tumor in relation to the wand. Brain-mapping techniques enable the surgeon to locate the vital areas controlling language and motor functions, helping the surgeon avoid that area during tumor removal. *Photodynamic therapy* (PDT) helps the neurosurgeon visualize the border of a tumor by use of a special dye injected prior to surgery. The dye accumulates in tumor cells, causing them to fluoresce when a laser is aimed at them. A laser is then used to vaporize the tumor cells. Microsurgical tools, the smallest of surgical instruments, combined with exquisitely high-powered microscopes are used to remove tumors in the base of the skull and/or around cranial nerves.

1.1.2 Radiation

Radiosurgery combines computer technology with radiation therapy to deliver focused, pin-point beams of high-dose radiation to the tumor. The goal of this approach is to limit the exposure of normal brain to the effects of radiation. Although there are three basic pieces of equipment used to deliver radiosurgery — the Gamma Knife, adapted linear accelerators, and cyclotrons which produce proton beams — the "brand names" of the equipment may vary with the manufacturer. For example, the Cyberknife and the X-knife are both linear accelerator-based radiosurgery systems. Conformal radiation, also called intensity modulated radiation therapy (IMRT), shapes the pattern of radiation beams to the shape of the tumor. Radioprotectors are drugs, such as DFMO, which protect brain cells during radiation therapy. Radiosensitizers appear to make tumor cells more sensitive to radiation. Both of these techniques may allow the use of higher, more effective doses of radiation with fewer radiation side-effects. Radioenhancers, such as RSR13 or Gd-texaphyrin, are designed to increase the efficiency of radiation therapy without increasing the dosage of radiation. Boron neutron capture therapy uses a boron compound activated by neutron radiation beams. Monoclonal antibodies are radioactive molecules used to deliver radiation or immunotherapy substances to a tumor.

The success of almost all these techniques depend on the accurate detection of brain tumor boundary. An accurate method for brain tumor boundary detection may lead to more effective brain surgeries. To make such sophisticated tools and techniques work properly, a range of methods including edge based, region based, knowledge based, and combination approaches have been proposed for semiautomatic and automatic detection of brain tumors and other structures in the head [1], [2]. Recently, several attempts have been made to apply neural network architectures, snakes based algorithms, curve evolution theory etc. to brain image analysis [3], [6], [8], [9]. Now we shall take a

brief look at these different existing techniques for medical image segmentation.

1.2 Existing Techniques

1.2.1 Anatomic Segmentation

The problem of segmentation is formulated as one of extracting substructure borders which are defined by regional intensity transitions [1]. The constructed boundary must be uninterrupted, and must circumferentially define the complete cross-section of a structure in an image section. As component structures merge with each other, an intensity transition zone of variable width is created. The width of the transition zone is dependent on the particular nature of the tissue. The general process of the segmentation is pursued in two ways. One method involves the classification and identification of all the pixels(picture elements) or voxels(volume elements) belonging to a specific structure e.g., a brain tumor. This involves the evaluation of the probability that any given pixel or voxel is a member of the desired region. Global intensity thresholding is a simple example of such a technique. This technique assumes that the probability for inclusion of a pixel or voxel in a region is directly proportional to that pixel/voxel's intensity. Thus, an intensity threshold is set such that all pixels or voxels above the threshold are included in the region, and those below the threshold are excluded. The second general method of segmentation involves the classification and identification of only the pixel or voxel locations constituting the border, or surface, of a given structure. A simple example of this is a feature of image intensity $\boldsymbol{A}$ against a background intensity $\boldsymbol{B}$. The boundary, or surface, of this region can then be defined as the pixels or voxels located at the $\boldsymbol{A}$ to $\boldsymbol{B}$ image intensity boundary.

1.2.2 Morphological Methods

Different morphological methods are widely used is various aspects of medical imaging [7], [13]. For segmentation purposes, a background image $\boldsymbol{B}$ is formed by gray scale opening of the original image $\boldsymbol{A}$. The background image $\boldsymbol{B}$ is subtracted from the original image to produce a less noisy image $\boldsymbol{C}$. This is contrast enhanced to get $\boldsymbol{D}$ and then thresholded to get $\boldsymbol{E}$. The separate foreground regions of the binary image are labelled and a "weight" is determined for each region. The weight of a region $\Re$ of the thresholded image $\boldsymbol{E}$ is calculated as follows. For each pixel p in the region $\Re$, the corresponding intensity of pixel p in the enhanced difference image $\boldsymbol{D}$ is added to a sum; this sum is defined as the weight of the region $\Re$. The weight is then a measure of both area and intensity and may be written as

$$W_{\Re} = \sum_{p \in \Re} D(p) \tag{1.1}$$

where $D(p)$ represents the intensity of the enhanced difference image $\boldsymbol{D}$ at the pixel location p. Depending on the application, the region of suitable weight is selected - for tumor detection, we select the region with the largest weight provided there were no larger, bright areas in the original image. After removing the unselected areas of the image, the selected region is filled in by performing a binary closing using a disc-like structuring element. Next a binary opening is performed to remove spurious extensions arising out of the closing operation. The resulting image is the segmented region of interest.

1.2.3 Active Contour Models

The snake model or the active contour model has been used extensively in medical image processing applications [4], [3], [9]. The basic model is an energy-optimizing

controlled continuity spline which can be manipulated by internal contour forces, image forces and external forces. The energy of a general snake is given by

$$E_{snake} = \int_{\Omega} (E_{int}(v) + E_{image}(v) + E_{ext}(v))ds \tag{1.2}$$

where $v \equiv v(s,t)$ is the deformable curve with parameters s(spatial index) and t(time index) defined respectively on given open intervals Ω and T. Initially the spline is put in the neighborhood of the desired energy-optima by the external forces. Once in the neighborhood of the optima, the image forces push the snake towards salient image features like lines, edges and subjective contours with a piecewise smoothness constraint imposed by the internal spline forces. A movement of the snake is allowed as long as the total energy is monotonically decreasing (increasing) if we want the snake to reach an energy minima (maxima). Finally, when the snake reaches the desired energy-minima (maxima), any change in the location or orientation of the snake results in an increase (decrease) in the total energy and so no further movement is allowed. This final position of the snake is the desired boundary.

1.2.4 Front Propagation

This is a modeling technique based on a level set approach for recovering shapes of objects in two and three dimensions [9], [10]. The modeling technique may be viewed as a form of active modeling such as snakes [4] with deformable surfaces. The model consists of a moving front which may be molded into any desired shape by externally applied halting criteria synthesized from the image data. The snakes or deformable surfaces may be viewed as Lagrangian geometric formulations where the boundary of the model is represented in a parametric form, as a front. Since the idea is to extract objects, shapes from a given image, the front should be forced to stop in the vicinity of the desired objects' boundaries. The speed function of the front is made to depend on

the inverse of the smoothened version of the gradient image. Since the gradient image has peaks at the boundaries and surfaces of the sub-structures in the image, the speed function will reduce and may actually become zero on approaching the boundaries. The final shape of the configuration is obtained when all the points on the front come to a stop, thereby bringing the computation to an end.

1.2.5 Neural Networks

Neural Networks are now being used for the segmentation and classification of magnetic resonance images. In many applications, Hopfield Neural Networks have been used as an energy optimizer for the crisp unsupervised classification and segmentation of MRI images [6], [3]. Hopfield networks minimizes an energy function of the form

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i \tag{1.3}$$

where N is the number of neurons, V_i is the output of the ith neuron, I_i is the bias term, and T_{ij} is the interconnection weight between the ith and jth neuron. If a problem can be cast in the form of or mapped to a minimization of Hopfield energy function,a neural network can be realized to obtain a solution. Recently two dimensional Hopfield Neural Networks [3] have been applied to the problem of brain tumor boundary detction.

In some of the methods like anatomic segmentation, morphological segmentation etc. only an approximate boundary can be determined, which is not good enough for applications related to the surgery of brain tumors. In most of the good methods employed for segmentation of brain tumor boundary, optimization of a proper energy function is involved. Now, the energy landscape in general has more than one minima (maxima) due to the nonconvex nature of the energy surface. The methods discussed so far can only assure the convergence to a local minima (maxima). In the active contour

model, the snake aligns itself to the local minima present in its neighborhood. In the case of front propagation, the moving front comes to a stop on the first local maxima it encounters while expanding or contracting. In case of Hopfield neural networks, a reasonably good solution (deep in the energy landscape) can be obtained, but it may actually be a local minima and once trapped, there is no way to get out of the valley, however shallow it may be.

All these observations have led us to approach the boundary detection problem with the help of the concepts involved in genetic algorithm. The standard genetic algorithm is a robust optimizer which works exceptionally well because of the random nature of the operations involved and that it deals with a population of possible solutions. Since genetic algorithm leads to the *survival of the fittest* under a given condition after some generations, the best fit solution emerges automatically after some generations, provided the *fineness function* has been properly formulated. Moreover, since the genetic algorithm essentially models a biological process of growth and evolution and the growth of a tumor is also a biological phenomenon, it is perhaps better suited to model the process. Before going into the details of the process of boundary detection, we shall take a brief look into the fundamental concepts involved in a genetic algorithm.

1.3 Genetic Algorithms: Basic Principles and Features

Genetic Algorithms(GAs) are intended to mimic some of the processes observed in natural evolution in the following ways [14]:

- Evolution is a process that operates on encoding of biological entities, rather

than on the living beings themselves, *i.e.*, evolution takes place at the level of chromosomes.

Similarly, GAs operate on encoding of possible solutions (called chromosomes) of the problems, by manipulating strings of characters of an alphabet.

- Natural selection is the link between a chromosome and its performance (measured on its decoded version). Nature obeys the principle of Darwinian "survival of the fittest"; the chromosomes with high fitness values will on the average, reproduce more often than those with low fitness values.

  In GAs, selection of chromosomes also tries to mimic this procedure. Highly fitted chromosomes will reproduce more often at the cost of lower fitted ones.

- Biological evolution has no memory. Here, nature acts as the environment and the biological entities are modified to adapt in their environment. Whatever nature knows about the evolution of good chromosomes is contained in the set of chromosomes possessed by the current individuals and in the chromosome decoding procedure.

  Likewise, GAs also operate on chromosomes blindly. They use only the *objective function* information which acts as the environment. Based on this information, each chromosome is evaluated and during the selection process more importance is given on choosing chromosomes having high fitness values.

- Like natural evolution, variation of the entities in GAs is introduced when reproduction occurs. Crossover and mutation are the basic operators of reproduction.

The GA-based evolution starts from a set of individuals (assumed solution set for the function to be optimized) and proceeds from generation to generation through genetic operations. Replacement of an old population with a new one is known as generation when the generational replacement technique (replace all the members of old population

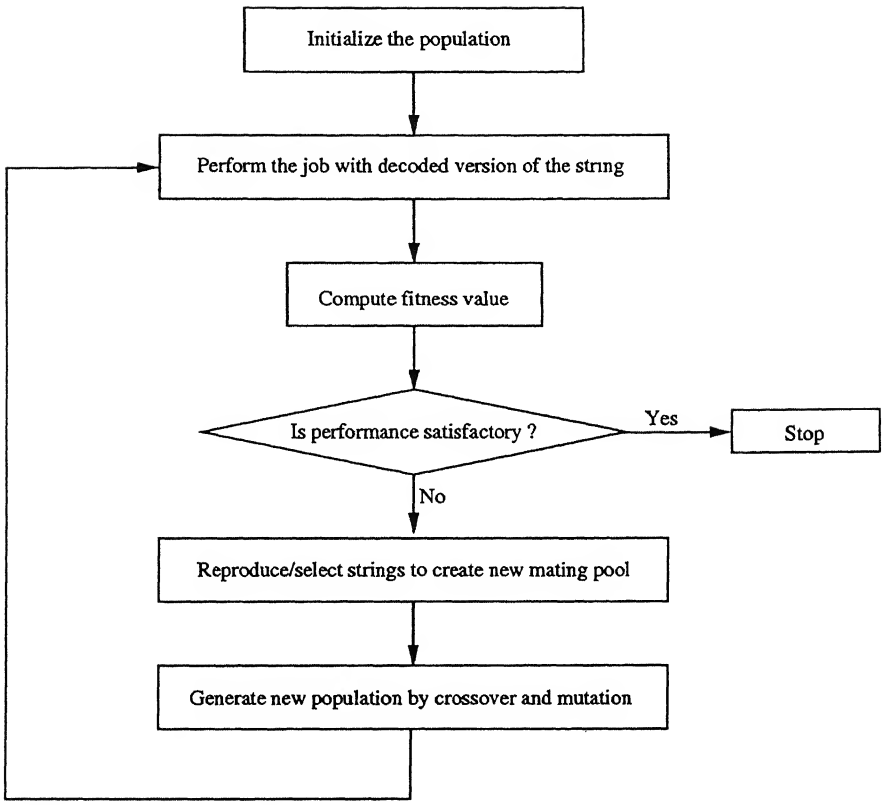


Figure 1.1: Basic steps of a genetic algorithm.

with the new ones) is used. Another reproduction technique, called steady-state reproduction, replaces one or more individuals at a time instead of the whole population. As mentioned before, GAs require only a suitable objective function which acts as the environment in order to evaluate the suitability of the derived solutions(chromosomes). A schematic diagram of the basic structure of a genetic algorithm is shown in Figure 1.1. A GA typically consists of the following components:

- A population of strings or coded possible solutions (biologically referred to as chromosomes)
- A mechanism to encode a possible solution (mostly as a binary string)
- Objective function and associated fitness evaluation techniques

- Selection or reproduction procedure
- Genetic operators
- Probabilities to perform genetic operations

Let us briefly describe these components:

1.3.1 Population

To solve an optimization problem, GAs start with structural (chromosome) representation of a parameter set $\{x_1, x_2, \ldots, x_p\}$. The parameter set is generally coded as a finite length string over an alphabet of finite length. Usually, the chromosomes are strings of 0's and 1's. For example, let $\{a_1, a_2, \ldots, a_p\}$ be a realization of the parameter set and the binary representation of $a_1, a_2, \ldots, a_p$ be $10110, 00100, \ldots, 11001$, respectively. Then the string $10110\,00100 \ldots 11001$ is a chromosome representation of the parameter set $\{a_1, a_2, \ldots, a_p\}$. It is evident that the number of possible chromosome (strings) is 2^l where, l is the string length. Each chromosome actually refers to a coded possible solution. A set of such chromosomes in a generation is called a population. The size of a population may vary from one generation to another or it can be constant. Usually, the initial population is chosen randomly.

1.3.2 Encoding and Decoding mechanism

It is the mechanism to convert the parameter values of a possible solution into binary strings resulting into chromosome representation. If the solution of a problem depend on p parameters and if we want to encode each parameter with a binary string of length

q, then the length of each chromosome will be $p \times q$. Decoding is just the reverse of encoding.

1.3.3 Objective function and associated fitness evaluation techniques

The fitness or objective function is chosen depending on the problem. It is chosen in a way such that highly fitted strings (possible solutions) have high fitness values. It is the only index to select a chromosome to reproduce for the next generation.

1.3.4 Selection or Reproduction procedure

The selection or reproduction process copies individual strings, called *parent chromosomes*, into a tentative new population, known as *mating pool*, for genetic operations. The number of copies reproduced for the next generation by an individual is expected to be directly proportional to its fitness value, thereby mimicking the natural selection procedure to some extent. Roulette wheel parent selection and linear selection are the most frequently used selection procedures.

Genetic operators are applied on parent chromosomes and new chromosomes (called offsprings) are generated. Frequently used genetic operators are described below.

1.3.5 Crossover

The main purpose of crossover is to exchange information between randomly selected parent chromosomes with the aim of not losing any important information (minimum

disruption of the structures of the chromosomes that are selected for genetic operations). Actually, it recombines genetic material of two parent chromosomes to produce offspring for the next generation. The general crossover operation can be viewed as a three step process. In the first step, pairs of chromosomes, called *mating pairs* are chosen from the mating pool. The second step determines, on the basis of crossover probability, whether these pairs of chromosomes should go for crossover or not. Interchange of chromosome segments between mating pairs is done in the third step. The number of segments and the length of each segment to be exchanged depend on the type of crossover technique. Some of the commonly used crossover techniques are one-point crossover, two-point crossover, multiple-point crossover, shuffle exchange crossover, uniform crossover etc. To illustrate how the segments of two parent chromosomes are swapped, let us consider the one-point crossover technique. Here, a position k is selected uniformly at random between 1 and $l-1$, where l is the length of the string (greater than 1). Two new strings are created by swapping all characters from position $(k+1)$ to l. Let

$$a = 11000\ 10101\ 01000 \ldots 01111\ 10001$$

$$b = 10001\ 01110\ 11101 \ldots 00110\ 10100$$

be two strings(parents) selected for the crossover operation and the generated random number in $< 0, l >$ be 11 (eleven). Then the newly produced offspring will be

$$a' = 11000\ 10101\ 01101 \ldots 00110\ 10100$$

$$b' = 10001\ 01110\ 11000 \ldots 01111\ 10001$$

1.3.6 Mutation

The main aim of mutation is to introduce genetic diversity into the population. Sometimes, it helps to regain information lost in earlier generations. Like natural genetic

systems, mutation in GAs is also made occasionally. A random position of a random string is selected and is replaced by another character from the alphabet. In case of binary representation it negates the bit value and is known as *bit mutation.* For example, let the third bit of string a, be selected for mutation. Then the transformed string after mutation will be

$$11100\ 10101\ 01000 \ldots 01111\ 10001$$

Random mutation is not always worth performing. High mutation rate can lead the genetic search to a random one. it may change the value of an important bit and thereby affect the fast convergence to a good solution. Moreover, it may slow down the process of convergence at the final stage of GAs.

1.3.7 Probabilities to perform genetic operations

The probability to perform crossover operation is chosen in a way so that recombination of potential strings increases without any disruption. Generally, the crossover probability lies inbetween 0.6-0.9.

Since mutation occurs occasionally, it is clear that the probability of performing mutation operation will be very low. Typically the value lies between 0.001 to 0.01.

Crossover and mutation probabilities may be kept fixed throughout the operation of a GA or they may be made adaptive to improve performance.

With this brief overview of the principles and concepts involved in a standard GA, we are now ready to take up the problem of brain tumor boundary detection. The rest of the thesis is organized as follows. In Chapter 2 we discuss the active contour model or the snake model and formulate the brain tumor boundary detection problem

as an optimization problem. The discrete form of the energy equation which is to be minimized is also derived. In Chapter 3 we propose an algorithm, based on GA concepts, to solve the optimization problem. Chapter 4 contains the results of applying the algorithm to MRI brain image slices. We have also discussed about the different aspects of the new algorithm and the scope for future work in the same chapter.

Chapter 2

The Active Contour Model

Active contour models, which originated from snakes [4], have been widely used in medical imaging processing recently. An active contour, which is commonly called *snake*, is an energy-minimizing spline generally guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes lock onto nearby edges, localizing them accurately.

The domain of problems that can be addressed using snakes consists of the task of finding salient image contours - edges, lines and subjective contours - as well as tracking those contours during motion and matching them in stereopsis. Unlike many other techniques for finding contours, this model is *active*. It is always minimizing its energy functional and therefore exhibits dynamic behavior. Because of the way the contours slither while minimizing their energy, we call them snakes. Snakes do not try to solve the entire problem of finding salient image contours. They rely on other mechanisms to place them somewhere near the desired contour. Often these external mechanisms are in the form of proper image preprocessing or by interaction with an expert user.

2.1 The Snake Energy Equation

Following the notation of Kass *et al.* [4], a deformable curve $v(s,t)$, with parameters s (spatial index) and t (time index), is defined on given open intervals Ω and T, respectively. By permitting the snake to have two deformational degrees of freedom in the plane, that is, in the x and y coordinates, the deformable active contour can be represented as

$$v(s,t) = (x(s,t), y(s,t)) : s \in \Omega, t \in T \tag{2.1}$$

The potential energy function of the snake can be written as

$$E_{snake} = \int_{\Omega} E(v)ds = \int_{\Omega} (E_{int}(v) + E_{image} + E_{ext}(v))ds \tag{2.2}$$

where E_{int} represents the internal energy of the spline due to bending, E_{image} is the image energy which gives rise to the image forces, and E_{ext} is any external energy in the form of external constraint forces.

2.1.1 The Internal Energy

The internal spline energy can be written as

$$E_{int} = \alpha(s)\|v'(s)\|^2 + \beta(s)\|v''(s)\|^2 \tag{2.3}$$

As we see, the energy is composed of a first-order term controlled by $\alpha(s)$ and a second-order term controlled by $\beta(s)$. According to Kass *et al.* [4], the first-order term makes the snake act like a membrane and the second-order term makes it act like a thin plate. Adjusting the weights $\alpha(s)$ and $\beta(s)$ controls the relative importance of the membrane and thin-plate terms. Zhu *et al.* [4] has interpreted $\alpha(s)$ and $\beta(s)$ to be regulating the elasticity and rigidity of the snake, respectively. In purely mathematical terms, they control the first and second order continuity of the snake. For example, setting $\beta(s)$ to

zero at a point allows the snake to become second-order discontinuous and develop a corner. For the application at hand and most other applications, discontinuities, both first and second order, are not desirable. If both $\alpha(s)$ and $\beta(s)$ are positive $\forall s$, a snake having any sort of discontinuity, sharp bent or corner will have a high value of internal energy and will not be able to minimize the energy equation in the given interval. Thus, the internal energy term ensures the 'smoothness' of the energy-minimizing spline.

2.1.2 The Image Energy

The image energy is chosen to depend on the gradient of the image since we want the contour to be attracted to edge points. The image energy is given by

$$E_{image} = -\gamma(s)\|\nabla(G_\sigma * I(x,y))\|^2 \tag{2.4}$$

where $\gamma(s)$ is the weight corresponding to the image energy term. The expression $(G_\sigma * I(x,y))$ denotes the image, $I(x,y)$, convolved with a gaussian smoothing filter whose characteristic width is σ. $\nabla(G_\sigma * I(x,y))$ is the gradient of the smoothed image. The gradient of the image is found using the Sobel operator [12].

For a continuous image $f(x,y)$, its derivative assumes a local maximum in the direction of the edge. The gradient of f along r in a direction θ with the x-axis is given by

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial r} = f_x \cos\theta + f_y \sin\theta \tag{2.5}$$

Since $\partial f/\partial r$ varies with θ, its maximum value is obtained when $(\partial/\partial\theta)(\partial f/\partial r) = 0$. This gives

$$-f_x \sin\theta_g + f_y \cos\theta_g = 0 \Rightarrow \theta_g = \tan^{-1}\left(\frac{f_y}{f_x}\right) \tag{2.6}$$

$$\left(\frac{\partial f}{\partial x}\right)_{max} = \sqrt{f_x^2 + f_y^2} \tag{2.7}$$

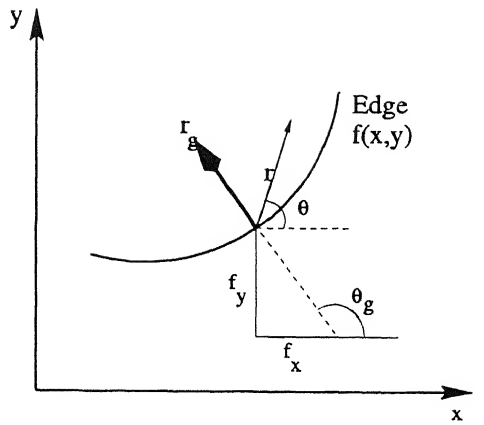


Figure 2.1: Gradient of $f(x, y)$ along direction r.

where θ_g is the direction of the edge. Based on these concepts, there are two basic type of edge detection operators - the *gradient operators* and the *compass operators.* For digital images these operators, also called *masks*, represent finite difference approximations of either the orthogonal gradients f_x, f_y or the directional gradient $\partial f/\partial r$. Let $\mathbf{H}$ denote a $p \times p$ mask and define, for an arbitrary image $\mathbf{U}$, their inner product at location (m, n) as the correlation

$$\langle \mathbf{U},\mathbf{H} \rangle_{m,n} \triangleq \sum_i \sum_j h(i,j)u(i+m, j+n) = u(m,n) * h(-m,-n) \tag{2.8}$$

Now, we shall concentrate on gradient operators. These operators are represented by a pair of masks $\mathbf{H_1}, \mathbf{H_2}$ which measure the gradient of the image in two orthogonal directions. Generally the bidirectional gradients are defined as $g_1(m,n) \triangleq \langle \mathbf{U}, \mathbf{H_1} \rangle_{m,n}$, $g_2(m,n) \triangleq \langle \mathbf{U}, \mathbf{H_2} \rangle_{m,n}$. The gradient vector magnitude and direction are given by

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)} \tag{2.9}$$

$$\theta_g(m,n) = \tan^{-1} \frac{g_2(m,n)}{g_1(m,n)} \tag{2.10}$$

In our case, we have used only the magnitude of the gradient, computed as

$$g(m,n) \triangleq \mid g_1(m,n) \mid + \mid g_2(m,n) \mid \tag{2.11}$$

A Sobel operator is a gradient operator which compute horizontal and vertical differences of local sums. This reduces the effect of noise in the data. The pair of masks for the Sobel operator are

$$\mathbf{H_1} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \mathbf{H_2} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.12}$$

We note that the operator has the desirable property of yielding zeros for uniform regions. We use this operator to find the gradient of the gaussian smoothed image, and use the gradient image for computing the image energy of the snake.

2.2 The Discretization

The discretization of the snake has to be done in the space domain for applying it to problems of image processing, since any image consists of pixels located at the lattice points of the image area. The spatial discretization is done by sampling the active contour v into N points $(v_i, i = 1, 2, \ldots, N)$, such that the contour is *connected.*

Definition 2.2.1 *For any two points u_1, u_2, where $u_i \equiv (x_i, y_i)$, u_1 is called a neighbor of u_2 if*

$$| x_1 - x_2 | \leq 1$$

and

$$| y_1 - y_2 | \leq 1$$

We denote u_1 *is a neighbor of* u_2 as $u_1 \rightsquigarrow u_2$. We note that $u_1 \rightsquigarrow u_2 \Rightarrow u_2 \rightsquigarrow u_1$.

Definition 2.2.2 *A contour v of N points $(v_i, i = 1, 2, \ldots, N)$ is called a connected contour if*

(a) $v_{i \bmod N} \rightsquigarrow v_{(i+1) \bmod N}$

(b) $\forall i, j,\ i \neq j,\ \exists$ *exactly two non-overlapping paths from* v_i *to* v_j *along the contour.*

The kth element of one of the paths joining v_i and v_j is $p_k = v_{(i+k-1) \bmod N}$, and that of the other path is $p'_k = v_{(i-k+1) \bmod N}$ where $k = 1, 2, \ldots$. The starting element p_1 of both the paths is v_i and the last element of both of them is v_j. Now that we have defined a connected contour, a contour from this point onwards shall mean a connected contour, unless otherwise specified. Moreover, all subscripts to v, x and y are modulo N.

Now that we have a discrete snake in the form of a contour, we go on to get the discrete form of the snake energy equation. The discretization of the internal energy is done by approximating the first and second order derivatives v' and v'' by finite differences. Before going into the discretization, we take a look at the definition of a *scaler point function.*

Definition 2.2.3 *Consider any region* $\Re$ *of space and suppose that to each point* $\mathbf{p}$ *of the same, there corresponds, by any law whatsoever, a scaler denoted by* $\phi(\mathbf{p})$. *Then* ϕ *is called a scaler point function defined for the region* $\Re$.

For a given contour, we consider the open interval Ω to be the region $\Re$. Then, corresponding to each 'point' s in the region, $\exists$ a scaler $v(s)$ and hence $v(s)$ is a scaler point function defined over Ω.

Definition 2.2.4 *The vector function* $\mathbf{i}\frac{\partial \phi}{\partial x} + \mathbf{j}\frac{\partial \phi}{\partial y}$ *is called the gradient of the scaler point function* $\phi : \Re \subseteq \mathbf{R^2} \rightarrow \mathbf{R}$ *and is denoted as* $\nabla \phi$.

At this stage we propose to replace the first and second order derivatives of $v(s)$ in the expression for the internal energy by the gradient, $\nabla v(s)$, and curvature, $\nabla^2 v(s)$, since $v(s)$ is a scaler point function. Then, the internal energy of the snake is given by

$$E_{int} = \alpha(s)\|\nabla v(s)\|^2 + \beta(s)\|\nabla^2 v(s)\|^2 \tag{2.13}$$

The justification of doing this follows.

Theorem 2.2.1 *Let the function* $\phi : \Re \subseteq \mathbf{R^2} \rightarrow \mathbf{R}$ *be differentiable at* $\mathbf{p} \in \Re$ *and let* $\nabla\phi(\mathbf{p}) \neq 0$. *Then*

(i) the directional derivative $(\partial\phi/\partial\mathbf{u})(\mathbf{p})$ *takes maximum value if and only if* $\mathbf{u}$ *is the unit vector in the direction* $\nabla\phi(\mathbf{p})$. *The maximum value is* $\|\nabla\phi(\mathbf{p})\|$

(ii) $(\partial\phi/\partial\mathbf{u})(\mathbf{p}) = 0$ *if and only if* $\mathbf{u}$ *is a unit vector orthogonal to* $\nabla\phi(\mathbf{p})$

Theorem 2.2.1 tells us that $\nabla\phi(\mathbf{p})$ is in the direction of maximum rate of change and is orthogonal to the direction of zero rate of change of ϕ at $\mathbf{p}$, which is very much desirable as a property of the gradient. The gradient takes on even more significance if we examine how $\nabla\phi(\mathbf{p})$ is related to the level set of ϕ that contains $\mathbf{p}$. Let the level set be $S = \{x \in \mathbf{R}^m \mid \phi(\mathbf{x}) = \phi(\mathbf{p})\}$. Since ϕ is constant in S, it is reasonable to expect that $(\partial\phi/\partial\mathbf{u})(\mathbf{p})$ will be zero when $\mathbf{u}$ is 'tangential' to S at $\mathbf{p}$. It would follow that $(\partial\phi/\partial\mathbf{u})(\mathbf{p})$ takes maximum value when $\mathbf{u}$ is a unit vector 'normal' to the level set S through $\mathbf{p}$. This in turn suggests that $\nabla\phi(\mathbf{p})$ is 'normal' to the level set S.

For a snake, a relatively smooth zone of the boundary will have a less value of the gradient and hence contribute a less value to the internal energy, whereas a zone with sharp bends will have higher values of the gradient and hence contribute more to the internal energy. Secondly, though we shall be taking absolute value of the gradient, the

direction of the gradient vector must be normal to the boundary of the snake. Now, keeping in mind the above two criterion, we propose two possible discretizations of the gradient operator and then decide which one is a better approximation.

Hypothesis 2.2.1

$$\|\nabla v(s)\|^2 \approx \|\nabla v_i\|^2 = \mid v_i - v_{i-1} \mid^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2$$

Here x_i and y_i are the x- and y-coordinates of the v_ith contour point, respectively. Since $\nabla^2 v_i = \nabla v_{i+1} - \nabla v_i$, we have

$$\begin{aligned} \|\nabla^2 v(s)\|^2 \approx \|\nabla^2 v_i\|^2 &= \mid v_{i-1} - 2v_i + v_{i+1} \mid^2 \\ &= (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \end{aligned}$$

Hypothesis 2.2.2

$$\|\nabla v(s)\|^2 \approx \|\nabla v_i\|^2 = \mid v_{i+1} - 2v_i + v_{i-1} \mid^2 = (x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2$$

In this case,

$$\begin{aligned} \|\nabla^2 v(s)\|^2 \approx \|\nabla^2 v_i\|^2 &= \mid v_{i+2} - 3v_{i+1} + 3v_i - v_{i-1} \mid^2 \\ &= (x_{i+2} - 3x_{i+1} + 3x_i - x_{i-1})^2 + (y_{i+2} - 3y_{i+1} + 3y_i - y_{i-1})^2 \end{aligned}$$

Now we shall consider a few cases in order to determine which one is a better approximation. Figure 2.2 shows some possible orientations of the pixels in the boundary of the snake. According to hypothesis 2.2.1, for the situations denoted in figures 2.2(a) and (b), the value of $\|\nabla v_i\|^2$ is 1 in both the cases, though the smoothness of the snake

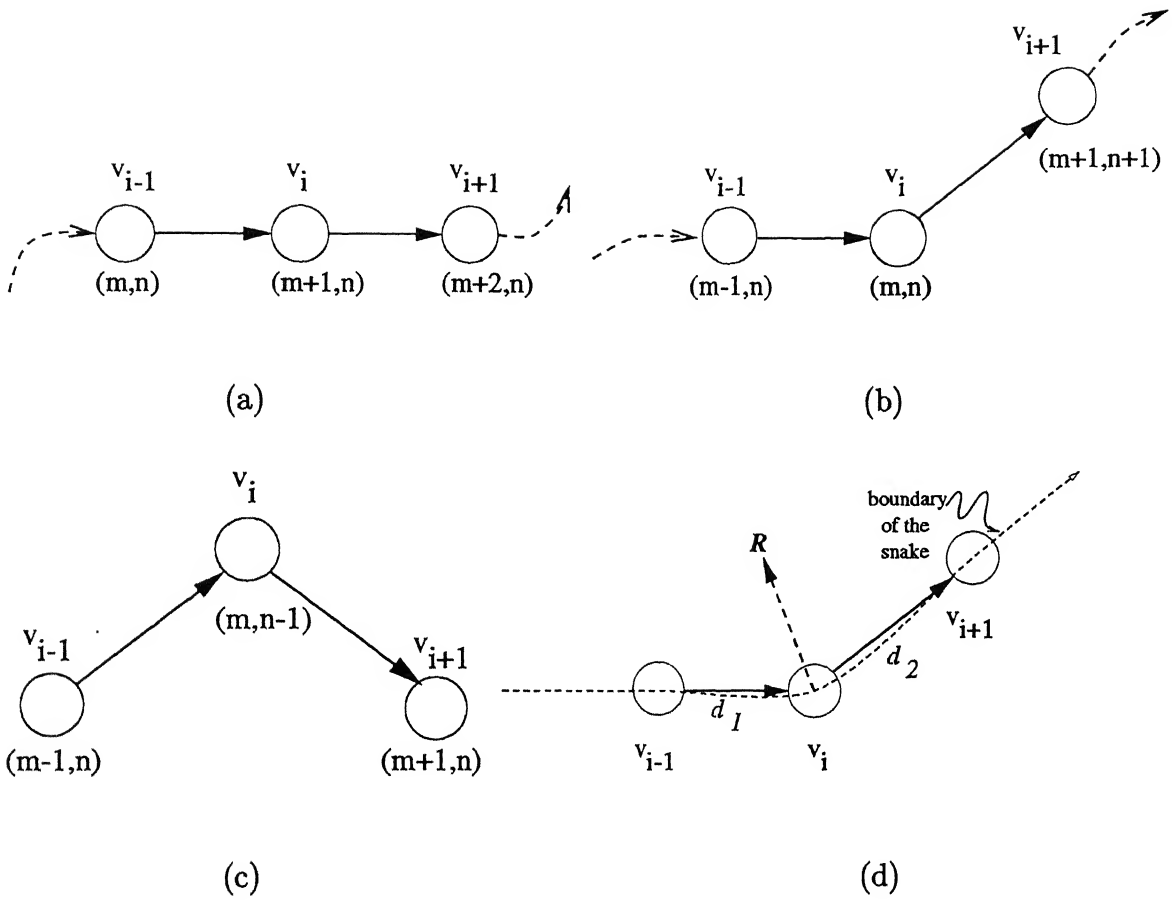


Figure 2.2: A few possible orientations of pixels v_{i-1}, v_i and v_{i+1} on the boundary of the snake - (a) a smooth boundary, (b) a slightly bent boundary, (c) a boundary with sharp bend, (d) the direction of the gradient vector.

varies distinctly in the two cases. Moreover, the value of $\|\nabla v_{i+1}\|^2$ in figures 2.2(b) and (c) are both equal to 2 though their amount of bending differ. Thus, the value of $\|\nabla v_i\|^2$ does not reflect the amount of bending of the snake at v_i. Also, the direction of ∇v_i is along d_1 as shown in figure 2.2(d) - we notice that it is not in the direction normal to the boundary of the snakes, rather it is along the boundary. The situation is a bit different if we use hypothesis 2.2.2. In that case, the value of $\|\nabla v_i\|^2$ for figure 2.2(a) is 0, thereby showing that the snake is perfectly smooth in this region; that for figure 2.2(b) is 1 and for figure 2.2(c) is 4. This reflects the fact that the more sharply the snake bends, the more will be the value of the gradient. Moreover, the direction of ∇v_i is along R (figure 2.2(d)), which is normal to the direction of the boundary of the snake, since :

$$\begin{aligned} \nabla v_i &= v_{i+1} - 2v_i + v_{i-1} = (v_{i+1} - v_i) - (v_i - v_{i-1}) \\ &= d_2 - d_1 = R \end{aligned}$$

So, we understand that hypothesis 2.2.2 is a better approximation and we shall use this discretization to find the discrete energy equation for a snake.

The discretization of the image energy is rather simple. The gradient of the gaussian smoothed image has a discrete gradient value $g(m, n)$ corresponding to all points (m, n) in the image. Let $g(i)$ be the gradient value corresponding to the ith point v_i of the snake under consideration. In other words, the point $v_i \equiv (x_i, y_i)$ of the gradient image has a value of $g(i)$. Since we do not threshold the gradient image after applying the Sobel operator, the value of $g(i)$ lies in the range [0-255]. Thus there will be a significant difference between a point having a gradient magnitude 240 and one having magnitude 255. To take care of this, we normalize the gradient magnitude of each contour point by

$$g_i = \frac{g(i)}{g_{max}} \tag{2.14}$$

where g_i is the normalized gradient magnitude, and g_{max} is the maximum gradient magnitude among the pixels located on the snake.

We do not have any explicit external constraint force acting on the snake and hence E_{ext} is always zero. In order to put the snake near the contour of interest, we use morphological pre-processing of the image, which is to be discussed in the next chapter. So, from the above discussion we conclude that the discrete form of the snake energy equation is given by

$$E_{snake} = \sum_{i=1}^{N} \{\alpha(i) \mid v_{i+1} - 2v_i + v_{i-1} \mid^2 + \beta(i) \mid v_{i+2} - 3v_{i+1} + 3v_i - v_{i-1} \mid^2 - \gamma(i) g_i^2\} \quad (2.15)$$

where $\alpha(i), \beta(i)$ and $\gamma(i)$ are the discrete versions of $\alpha(s), \beta(s)$ and $\gamma(s)$. In our algorithm we have used fixed values, α, β and γ, for these weights. We note that this energy equation has a dependence on the N, the length of the snake, especially the terms corresponding to the internal energy of the snake. Since all the terms corresponding to the internal energy are non-negative, a higher value of N will always give a higher value of the internal energy. Thus a bigger circular snake will always have a higher internal energy than a smaller circular snake. Since we measure the 'amount of bending' in a snake by means of the internal energy, and the 'amount of bending' in two circles, whatever their size may be, is intuitively the same on the whole, the two circles should have the same internal energy. Moreover, since a longer snake tends to have higher internal energy, other things remaining the same, there will be a tendency of the snake to shrink while the minimization process is going on, which may not be desirable. To tackle this problem, we take some sort of an average. We divide the energy Eqn. 2.15 by N, thereby doing away with the dependence of the energy on the length of the snake. Another constraint we would like to impose is that E_{snake} should always be positive. Since g_i is normalized, we have

$$1 \geq g_i^2 \geq 0 \quad \Rightarrow \quad -\gamma \leq -\gamma g_i^2 \leq 0$$

Since E_{int} is always non-negative we can say that $E_{int} + E_{image} \geq -\gamma$. Thus if we add a fixed offset γ to our existing energy equation, we shall always have a non-negative snake energy. Keeping all this in mind the final form the snake energy is

$$\begin{aligned} E_{snake} &= \frac{1}{N}\sum_{i=1}^{N}\{\alpha \mid v_{i+1} - 2v_i + v_{i-1} \mid^2 + \beta \mid v_{i+2} - 3v_{i+1} + 3v_i - v_{i-1} \mid^2 \\ &\quad -\gamma(g_i^2 - 1)\} \qquad (2.16) \\ &= \frac{1}{N}\sum_{i=1}^{N}\{\alpha[(x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2] \\ &\quad +\beta[(x_{i+2} - 3x_{i+1} + 3x_i - x_{i-1})^2 + (y_{i+2} - 3y_{i+1} + 3y_i - y_{i-1})^2] \\ &\quad -\gamma(g_i^2 - 1)\} \qquad (2.17) \end{aligned}$$

The detection of the exact boundary of a brain tumor can be done by minimizing the above energy function. The contour or snake that minimizes the above energy equation is the desired boundary. In the next chapter we shall describe, in detail, how this minimization is to be done.

Chapter 3

The Optimization Algorithm

In this chapter we shall describe a new algorithm based on GA concepts for the minimization of the energy equation 2.17. Though the algorithm uses the concepts involved in a GA, the implementation of these ideas has been done in quite a non-standard fashion. It has been observed [14] that ordinary linear genetic algorithms do not appear to function optimally in the framework of image processing. So we have proposed a *snake-based genetic algorithm.*

3.1 An Overview

This boundary detection approach has been designed to operate on multislice MRI scan of the human brain to extract the boundaries of brain tumor in each slice. As an input, we have a number of MRI slices of a brain having a tumor located somewhere in it (Fig. 3.1). We shall briefly describe how the boundary is detected in each slice. In the beginning we do some sort of morphological preprocessing on the given image to get

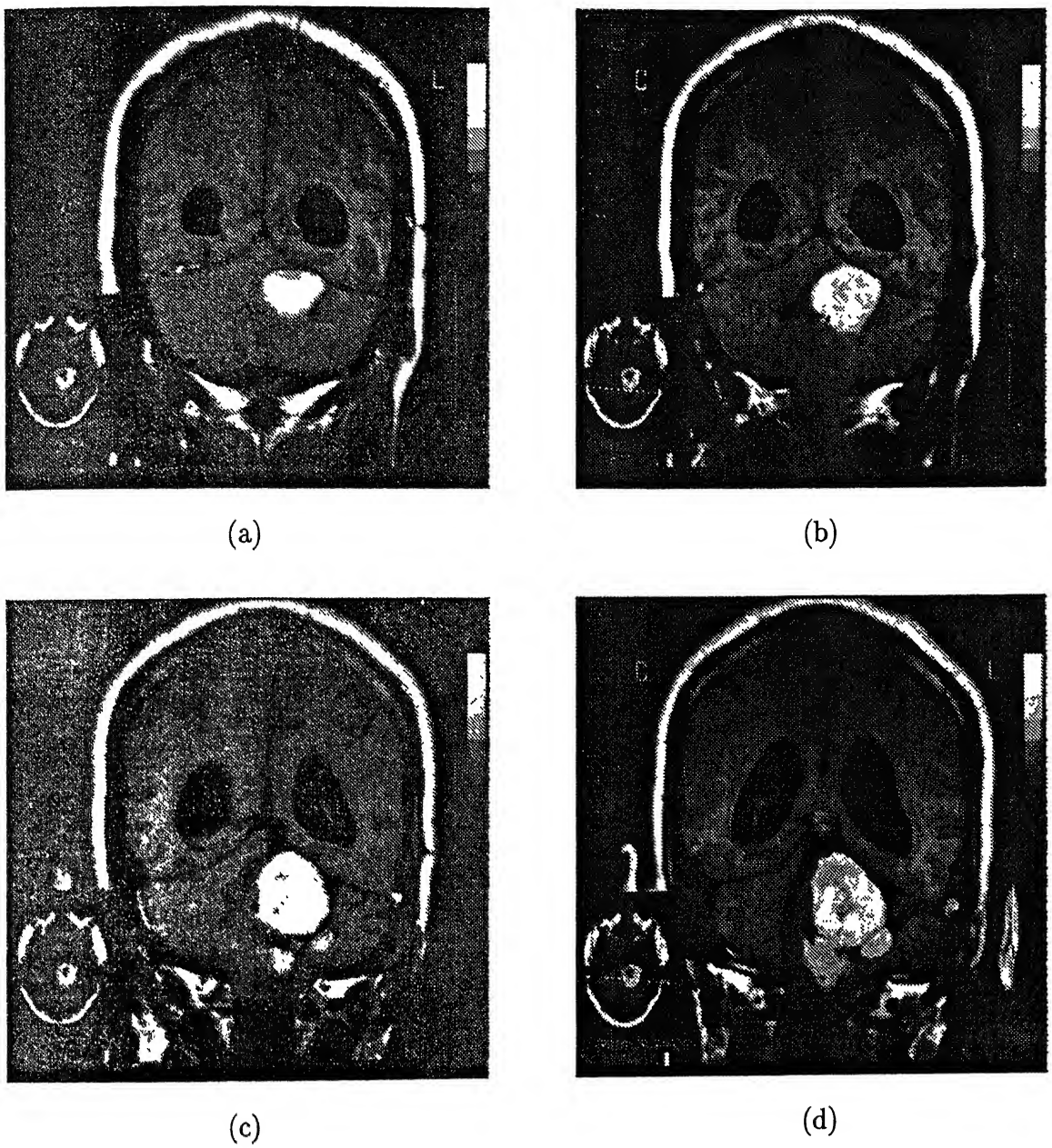


Figure 3.1: Example image slices from an axial MRI data set containing brain tumor.

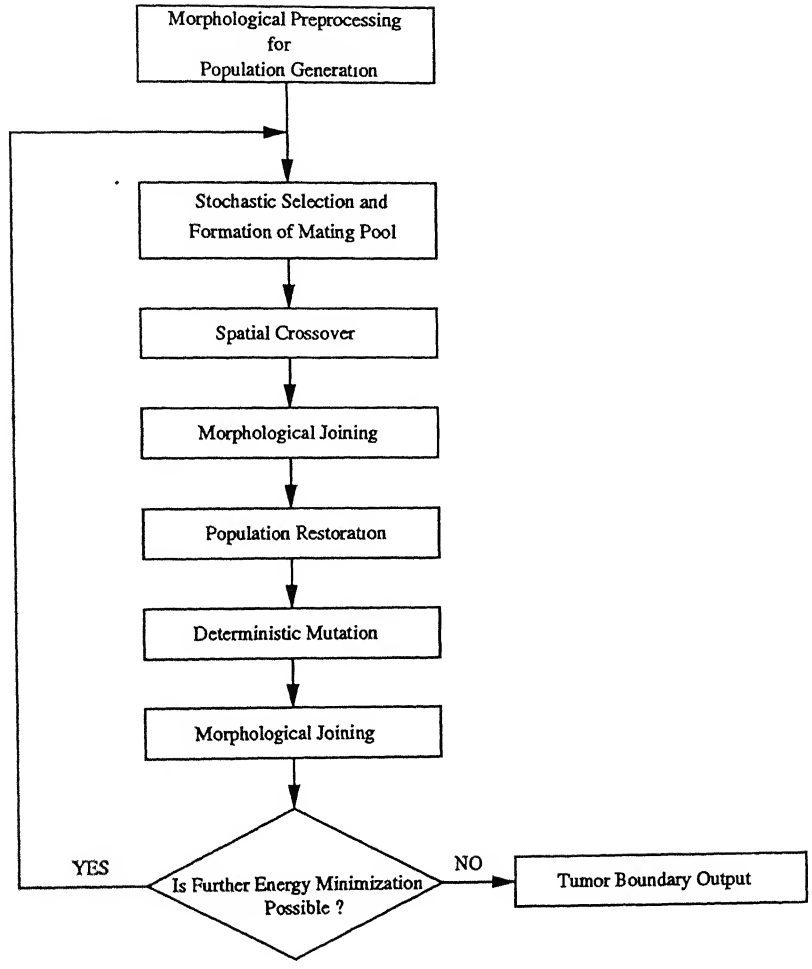


Figure 3.2: Block diagram of the proposed brain tumor detection algorithm.

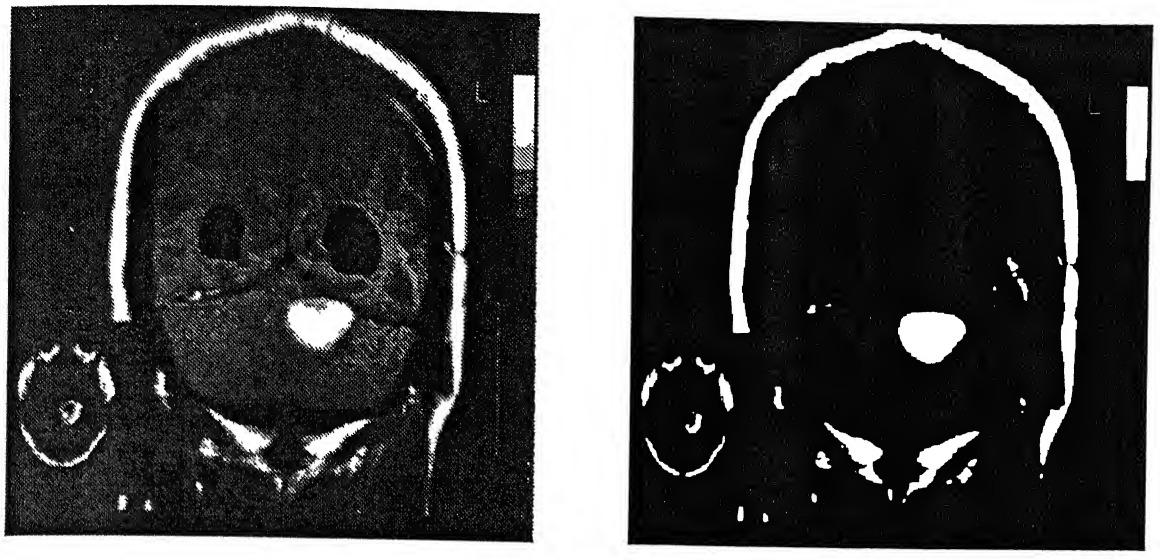
a *population* of eligible *boundary contours* (BCs). With the help of the morphological operations and the use of different morphing masks, regions approximating the actual tumor are obtained. The edge-detected and thinned versions of the boundaries of these regions are the BCs. These BCs are in fact different approximations of the true boundary. Next, we extract the *connected contours* from the thinned versions of the population of BCs, thereby yielding *connected boundary contours* (CBCs). It is to be noted that standard thinning algorithms do not give a connected contour according to the description in Eqn. 2.2.2.

A part of the population of CBCs go into the mating-pool for *reproduction* depending upon their *fitness*, which once again depends on the value of the energy function computed along the contour of the CBCs. Reproduction takes place randomly between any pair of the CBCs present in the mating pool. A CBC in the mating pool can go in for reproduction only once in a particular *generation*. During reproduction between two CBCs, spatial *crossover* takes place giving rise to a single child. This child may or may not be a BC, let alone a CBC. A *joining* operation, which is something similar to the morphological closing, is done on the child to form a BC and eventually a CBC. In the whole process, we want to keep the population of the CBCs constant. If we start off with an initial population of P CBCs, after the crossover and joining we shall have $(P + P/2)$ CBCs. P CBCs are chosen randomly out of this group and these go in for *mutation*. In contrary to standard GAs, the mutation operation is very deterministic and involved in our case. The CBCs formed after mutation are morphologically joined to form smooth BCs and eventually CBCs. This completes one generation of computations. The basic structure of the algorithm is shown in figure 3.2.

3.2 Morphological Preprocessing

We first *threshold* the gray level image to extract the tumor. Assuming the brain tumor has higher gray level values (Fig. 3.1) than its surrounding tissues, the threshold value is set to a level above the white matter. Actually, the threshold value is not crucial to the boundary detection since only a rough boundary is needed at this stage. For an image $I(m, n)$, the thresholded image is given by

$$\Theta_t(m,n) = \begin{cases} 1, & (m,n) \in I_t \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$



(a) (b)

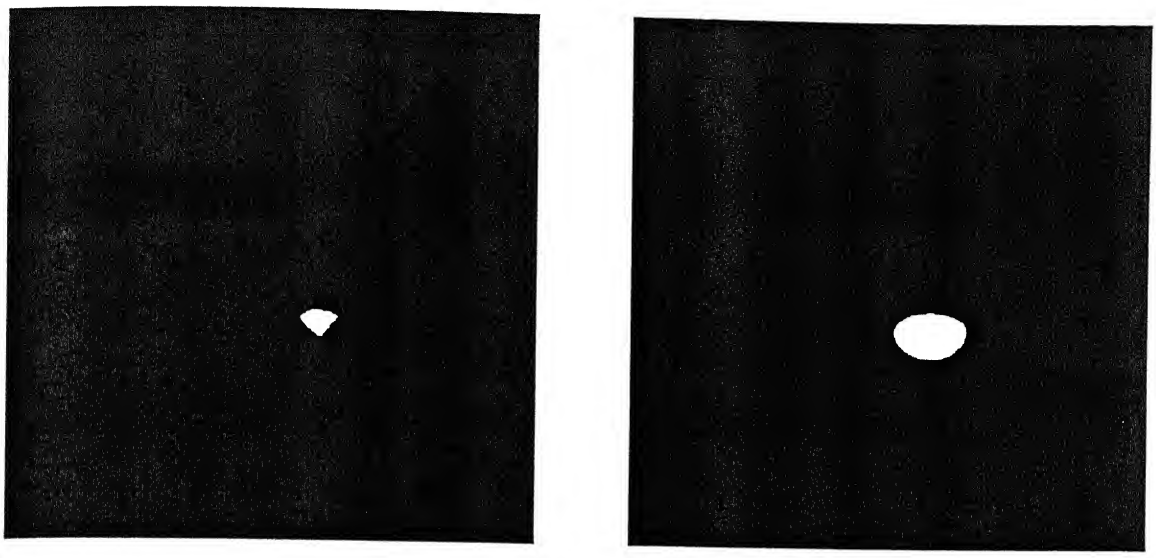
Figure 3.3: (a)The gray level image containing brain tumor, (b) Corresponding image after thresholding.

where $I_t \triangleq \{(m,n) \mid I(m,n) > t\}$. A gray scale image and its corresponding thresholded image is shown in figure 3.3. As can be seen from the figure, the resulting thresholded image Θ_t includes a tumor, some tissues with intensities as high as the tumor pixels and some noisy structures. To extract the tumor, a binary morphological erosion and dilation process [11] is applied to the thresholded image Θ_t to remove noise as well as normal tissue structures while retaining the brain tumor areas.

For *binary erosion*, a square structuring element, SE_ϵ, of appropriate size is used so that all the unwanted tissues and noise in the thresholded image is removed. For a thresholded image Θ_t, the eroded image

$$\mathcal{E} = (\Theta_t \ominus SE_\epsilon) \tag{3.2}$$

Fig. 3.4(a) shows how erosion removes the spurious regions whose areas are significantly smaller than that of the tumor.



(a) (b)

Figure 3.4: (a)The eroded version of the thresholded image in Fig. 3.3(a), (b)The eroded image after dilation using a disc shaped structuring element.

After this, a proper *dilation* is performed on the eroded image $\mathcal{E}$ using smooth structuring elements, SE_δ, so that the part of the tumor which was removed during erosion is approximately restored. The dilated image $\mathcal{D}$ obtained after dilating $\mathcal{E}$ by a structuring element SE_δ can be expressed as

$$\mathcal{D} = (\mathcal{E} \oplus \mathcal{SE}_\delta) \tag{3.3}$$

The smooth structuring elements used for the purpose may be disc-shaped, polygonal with smooth edges, circular and even tumor-shaped. A number of these structuring elements are used on the same eroded image to get a population of dilated images approximating the tumor. The justification of using smooth structuring elements is that it would not give sharp edged dilated image as an approximation of the actual tumor. Fig. 3.4(b) displays the extracted approximate tumor after dilation using a disc shaped structuring element.

The boundary of the approximate tumor region is obtained by using standard Sobel

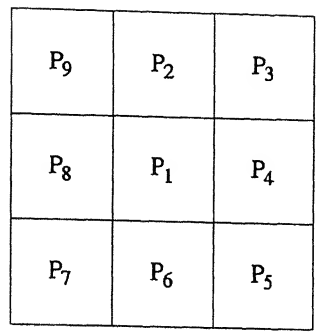


Figure 3.5: Neighborhood arrangement used by the thinning algorithm.

operators [12]. The boundary extracted by applying Sobel operator is generally a few pixels thick and is hence not suitable for CBC extraction. *Thinning* of this boundary is necessary before CBC extraction and is brought about by the following algorithm [13].

3.3 The Thinning Algorithm

In this discussion it is assumed that region points have value 1 and background points have value 0. The method consists of successive passes of two basic steps applied to the contour points of the image, where a *contour point* is any pixel with value 1 and having at least one 8-neighbor valued 0. With reference to the 8-neighborhood definition shown in Figure 3.5, we can say that p_1 is a contour point if any one of $p_i, i = 2, 3, \ldots, 9$ is 0. The first step flags a contour point p_1 for deletion if the following conditions are satisfied :

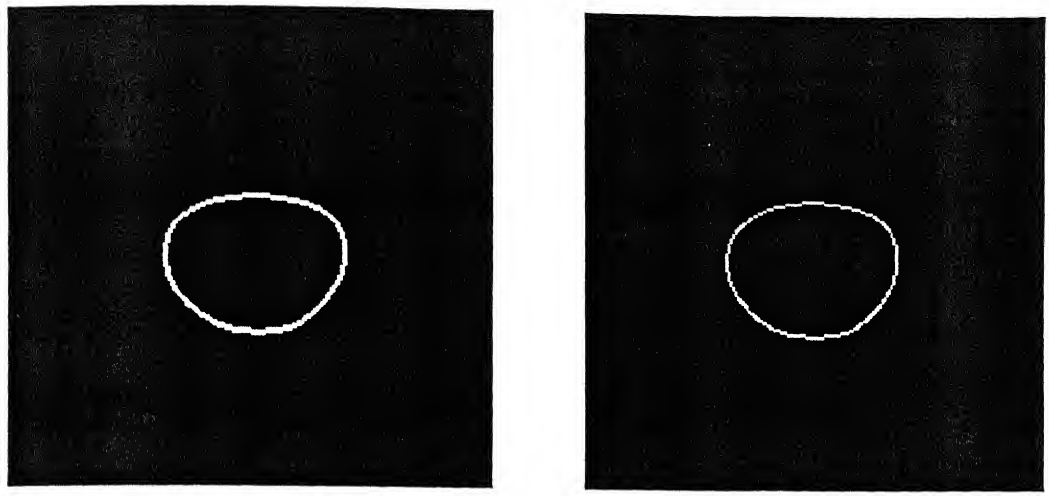
$$\begin{aligned} &(a) \quad 2 \leq N(p_1) \leq 6, \\ &(b) \quad S(p_1) = 1, \\ &(c) \quad p_2 \cdot p_4 \cdot p_6 = 0, \\ &(d) \quad p_4 \cdot p_6 \cdot p_8 = 0, \end{aligned} \tag{3.4}$$

where $N(p_1)$ is the number of nonzero neighbors of p_1, that is $N(p_1) = \sum_{i=2}^{9} p_i$ and $S(p_1)$ is the number of 0-1 transitions in the ordered sequence of $p_2, p_3, \ldots, p_9$. In the second step, conditions (a) and (b) remain the same, but conditions (c) and (d) are changed to

$$\begin{aligned} &(c') \quad p_2 \cdot p_4 \cdot p_8 = 0, \\ &(d') \quad p_2 \cdot p_6 \cdot p_8 = 0, \end{aligned} \tag{3.5}$$

Step 1 is applied to every border pixel in the binary region under consideration. If one or more of the conditions (a) through (d) are violated, the value of the point in question is not changed. If all conditions are satisfied the point is flagged for deletion. It is important to note that the point is not deleted until all border points have been processed. This prevents changing the structure of the data during execution of the algorithm. After Step 1 has been applied to all border points, those that were flagged are deleted (i.e.,changed to 0). Then Step 2 is applied to the resulting data in exactly the same manner as step 1. This basic procedure is applied iteratively until no further points are deleted, at which time the algorithm terminates yielding the skeleton of the region.

Condition (a) is violated if the contour point p_1 is an isolated point having all the 8-neighbors valued 0 in which case it remains unchanged, or if p_1 has only one or seven 8-neighbors valued 1. Having only one such neighbor implies p_1 is the end point of a skeleton stroke and obviously should not be deleted. If p_1 has seven such neighbors and it was deleted, this would cause erosion into the region. Condition (b) is violated when it is applied to intermediate points on a stroke one pixel thick. Thus this condition prevents disconnection of segments of a skeleton during the thinning operation. Conditions (c) and (d) are satisfied simultaneously by the following minimum set of values: $p_4 = 0$, or $p_6 = 0$, or ($p_2 = 0$ and $p_8 = 0$). Thus with reference to the neighborhood arrangement in Fig. 3.5, a point that satisfies these points as well as conditions (a) and (b) is an east or south boundary point or a northwest corner point in the bound-



(a) (b)

Figure 3.6: (a) The edge-detected version of the approximate tumor region in Fig. 3.4(b), (b) The BC obtained from (a) after applying the thinning algorithm. Both the images have been zoomed to get a better view.

ary. In either case p_1 is not a part of the skeleton and should be removed. Similarly, conditions (c') and (d') are satisfied simultaneously by the following minimum set of values : $p_2 = 0$, or $p_8 = 0$, or ($p_4 = 0$ and $p_6 = 0$). These correspond to north or west boundary points, or a southeast corner point. It is to be noted that northeast corner points have $p_2 = 0$ and $p_4 = 0$ and thus satisfy conditions (c) and (d), as well as (c') and (d'). This also true for southwest corner points, which have $p_6 = 0$ and $p_8 = 0$. Thus all the four boundaries and corners of the thick image get affected by the algorithm which eventually gives rise to the skeleton. Fig. 3.6(b) shows the result of applying the thinning algorithm to a thick contour(Fig. 3.6(a)) obtained after edge detection of one of the approximated tumor regions (Fig. 3.4(b)).

The process is repeated for all the approximate regions obtained from dilation. As a result, we get a population of thinned BCs from which the CBCs are to be extracted.

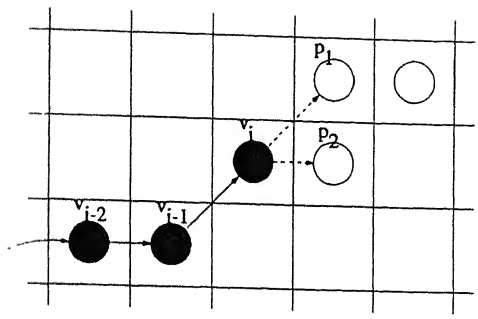


Figure 3.7: Alternative pixels p_1 and p_2 for selecting the next node v_{i+1}.

3.4 Contour Extraction

The contour extraction problem can be formulated as one of minimum cost tree searching and is carried out in this case using a heuristic search algorithm [5]. The formulation of the contour extraction problem as a tree searching problem hinges on the following factors. First, it is necessary that a root node be properly defined. We shall consider the contour to be extracted as a directed closed graph consisting of a set $\{v_i\}$ of nodes and a set $\{e(i, i+1)\}$ of edges, with each node having eight nearest neighbors. For any image, we choose the root, v_1, to be the top-left-most pixel of the thinned BC. Second, from among the eight nearest neighbors of a node v_i already identified to be lying on the contour, we shall have to select the next node v_{i+1} from the *on* points, i.e., the points already lying in the thinned BC, obviously excluding the predecessor v_{i-1} of the node under consideration. Fig. 3.7 shows a situation where we have two options in selecting v_{i+1}. Third, since we want to a closed CBC, we must have the constraint that if there a total of N nodes in the graph, $v_N \rightsquigarrow v_1$.

The extraction of the contour is done on the basis of a principle of *minimum deviation*, as first and second order derivatives in the energy function to be evaluated requires *smoothness* of the extracted CBC. We shall define the cost associated with a particular edge of the graph based on this principle. The cost, $\delta(i, i+1)$, of a particular

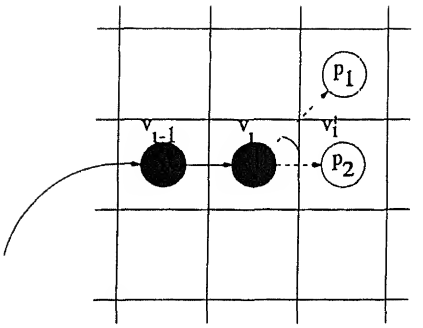


Figure 3.8· Scheme for cost estimation of an edge.

edge $e(i, i+1)$ depends on the angle by which the head of a snake moving along the contour has to move to reach from node v_i to node v_{i+1}. If v_i' be the node the snake would have reached from v_i without any head movement, then we define

$$\delta(v_i, v_{i+1}) = \frac{4}{\pi} \mid \angle(v_{i+1}, v_i, v_i') \mid \tag{3.6}$$

From Fig. 3.8, if we test the eligibility of p_1 for being the next node v_{i+1}, we have $\mid \angle(v_{i+1}, v_i, v_i') \mid = \frac{\pi}{4}$ and hence $\delta(i, i+1) = 1$ In case of p_2, $\delta(i, i+1) = 0$ since there is no deviation required for reaching p_2 from v_i.

A reasonable performance measure is the cumulative cost associated with a specified contour $\mathcal{V}$ and can be expressed as

$$\mathcal{C}(\mathcal{V}) = \sum_{i=2}^{N} \delta(v_i, v_{i+1}) \tag{3 7}$$

where N, the total number of nodes, is not known when we start the process of computing the cumulative cost. We note that the initial edge $e(1, 2)$ consisting of the root v_1 as well as its immediate successor v_2 need to be fixed from beforehand for proper evaluation of costs corresponding to the other edges. Since $v_1 \equiv (x_1, y_1)$ has been fixed already we choose $v_2 \equiv (x_2, y_2)$ to be the node such that $[x_1 - x_2 - y_1 - y_2]$ is a minimum. So, the problem of contour extraction is essentially that of finding $\mathcal{V}$ such that $\mathcal{C}(\mathcal{V})$ is minimized. Now, from Bellman's *principle of optimality* [12], the optimum

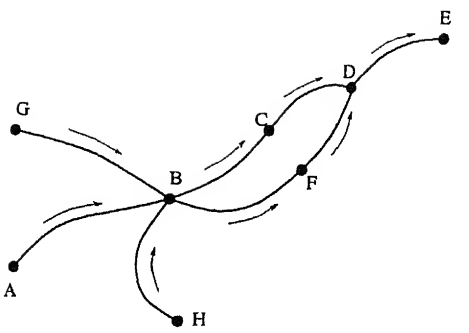


Figure 3 9: Bellman's principle of optimality. If the path ABCDE is optimum, then so is BCD no matter how you arrive at B.

path between any two given nodes is also optimum between any two points lying on the path. Thus if B is a point on the optimum path $ABCDE$ between A and E (Fig. 3 9), then the segment BCD, and not BFD, is the optimum path from B to D, no matter how one arrives at B Following this principle, starting from the node v_2, we go on finding the nearest neighbor v_{i+1} of $v_i, i = 2, 3, \ldots$, such that the cost of going from v_i to v_{i+1} is minimum. The process continues until for a node v_i under consideration, the neighbor satisfying the criterion is v_1. The process of contour extraction comes to an end here. This process forms a very fundamental part of the entire algorithm since all snake energy computations, the crossover and the mutation operations are preceded by this contour extraction in order to get a CBC from the BC

3.5 The Selection

Let there be P CBCs in the initial population. It is to be noted that our algorithm works with a fixed population and hence the population of the CBCs will be P through all generations. So, we have a set of P CBCs $\mathcal{V}_k, k = 1, 2, \ldots, P$. The discrete snake energy $E_{snake}(\mathcal{V}_k)$ of each of the P extracted CBCs is computed using the Eqn 2.17.

Now, we define the *fitness* of the kth CBC $\mathcal{V}_k$ as

$$\mathcal{F}(\mathcal{V}_k) \triangleq \frac{\mathcal{G}}{\mathcal{B} + E_{snake}(\mathcal{V}_k)} \tag{3.8}$$

where $\mathcal{G}$ and $\mathcal{B}$ are the *gain* and *base* of the fitness function. The value of $\mathcal{G}$ and $\mathcal{B}$ are design criterion and was set to 10 0 and 1.0 respectively after some experimentation

After computation of the fitness of all the P CBCs, we are now in a position to select P CBCs, obviously with repetitions, for the formation of the mating pool. Initially we make a *stochastic remainder* selection of the CBCs rather than a *biased roulette wheel* selection since the later is supposed to be more noisy. At this point we define $\mathcal{R}_{int}(x)$ to the function that rounds x to the nearest integer. At first $\mathcal{R}_{int}(\mathcal{F}(\mathcal{V}_k)/\mathcal{G}), \forall k$ samples of the CBC $\mathcal{V}_k$ are entered into the mating pool. Since we ensure a non-negative snake energy $E_{snake}(\mathcal{V}_k), \forall k$, the minimum value of $E_{snake}(\mathcal{V}_k)$ can be 0 and hence the maximum value of $\mathcal{F}(\mathcal{V}_k)$ can be $\mathcal{G}/\mathcal{B}$. Hence for any CBC $\mathcal{V}_k$, at most $\mathcal{R}_{int}(\frac{\mathcal{F}(\mathcal{V}_k)}{\mathcal{G}}) = \mathcal{R}_{int}(\frac{\mathcal{G}/\mathcal{B}}{\mathcal{G}}) = \mathcal{R}_{int}(\frac{1}{\mathcal{B}}) = 1$ sample of $\mathcal{V}_k$ can enter the mating pool as $\mathcal{B} = 1\ 0$ If a total of P_1 CBCs enter the mating pool by this method, the remaining $P_2 = (P - P_1)$ CBCs are selected using the biased roulette wheel principle.

In the biased roulette wheel principle, each CBC in the population has a slot size proportional to its fitness(Fig. 3 10). Theoretically, the principle is as follows. We have a fixed pointer S. The roulette wheel is put into rotation. As it slows down and stops, the pointer S will be lying in one of the P unequal sized slots. If S points to the kth slot, the corresponding CBC $\mathcal{V}_k$ is selected. The mathematical formulation of the same is as follows. We define a *cumulative fitness* corresponding to each CBC $\mathcal{V}_k$ given by

$$\mathcal{C}_k = \mathcal{C}_{k-1} + \frac{\mathcal{F}(\mathcal{V}_k)}{\sum_{h=1}^{P} \mathcal{F}(\mathcal{V}_k)} \tag{3.9}$$

where $\mathcal{C}_0 = 0$. Using a random number generator to produce r, $0 < r \leq 1$, if $\mathcal{C}_{k-1} < r < \mathcal{C}_k$, then a sample of the kth CBC $\mathcal{V}_k$ is selected and passed on to the mating pool.

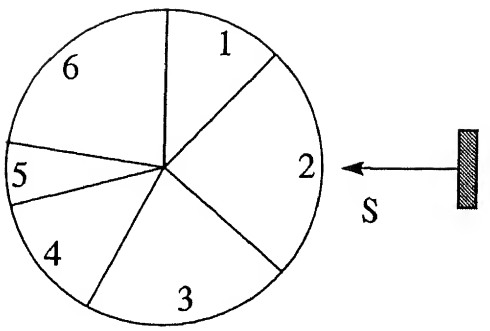


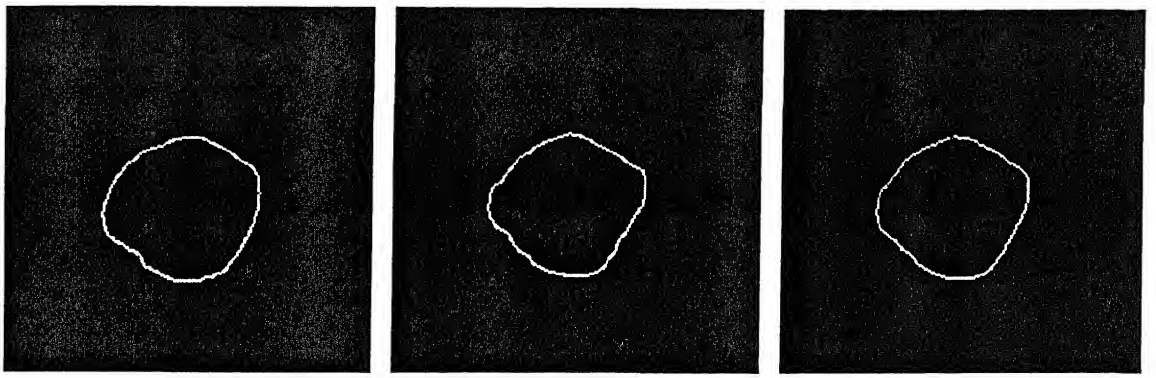
Figure 3.10· The roulette wheel principle. If the wheel stops rotating in this position, $\mathcal{V}_2$ is selected.

This process is repeated P_2 times after which we shall have $P_1 + P_2 = P$ selected CBCs in the mating pool.

3.6 The Crossover

The crossover operator implements a two-dimensional spatial averaging of the two CBCs involved in the crossover. The properties of this operator are as follows. If there are odd number of CBCs in the mating pool, one of the CBCs does not take part in the crossover operation First, all the CBCs present in the mating pool after selection take part in the crossover exactly once Second, the crossover partner for any CBC in the pool is selected randomly. Third, the crossover of two *parent* CBCs give rise to exactly one *child* CBC. Thus, for a population of P CBCs, there shall be a total of $\mathcal{R}_{int}(\frac{(P-1)}{2})$ crossovers. After all the crossovers, there will be $\mathcal{R}_{int}(P + \frac{(P-1)}{2})$ CBCs in the mating pool of which P are the parent CBCs and the rest are new-borns.

The basic crossover operation between two CBCs $\mathcal{V}_k$ and $\mathcal{V}_l$ is as follows. Let N_k be the number of nodes of the CBC $\mathcal{V}_k$ and let N_l be that of the CBC $\mathcal{V}_l$ Then the child $\mathcal{V}_{ch}$ will initially have $(N_k + N_l)/2$ nodes. Thus the ratio of the number of nodes



(a) (b) (c)

Figure 3 11: The crossover operation The spatial averaging or crossover of the parents (a) and (b) give birth to their child (c).

of the child $\mathcal{V}_{ch}$ and the two parents $\mathcal{V}_k$ and $\mathcal{V}_l$ respectively is

$$\frac{N_k+N_l}{2} : N_k : N_l \equiv 1 \quad \frac{N_k}{\frac{N_k+N_l}{2}} \cdot \frac{N_l}{\frac{N_k+N_l}{2}} \equiv 1 \cdot \frac{2N_k}{N_k+N_l} \cdot \frac{2N_l}{N_k+N_l} \tag{3 10}$$

So, corresponding to 1 node in the child, we have $\frac{2N_k}{N_k+N_l}$ nodes in$\mathcal{V}_k$ and $\frac{2N_l}{N_k+N_l}$ nodes in $\mathcal{V}_l$ Keeping this in mind, we get the mth point of the child as

$$\begin{aligned} v_m^{ch} &\equiv (x^{ch}(m), y^{ch}(m)) \\ &= \left(\frac{x^k\left(\mathcal{R}_{int}(\frac{2mN_k}{N_k+N_l})\right) + x^l\left(\mathcal{R}_{int}(\frac{2mN_l}{N_k+N_l})\right)}{2}, \frac{y^k\left(\mathcal{R}_{int}(\frac{2mN_k}{N_k+N_l})\right) + y^l\left(\mathcal{R}_{int}(\frac{2mN_l}{N_k+N_l})\right)}{2} \right) \end{aligned} \tag{3.11}$$

This is essentially a spatial averaging operation. Varying m from 1 to $\frac{N_k+N_l}{2}$, we get the initial form of the child $\mathcal{V}_{ch}$ (Fig. 3.11).

The above method of spatial averaging may lead to a contour which has some discontinuities in its boundary In order to get a continuous contour, we perform something similar to a morphological closing operation which we call *joining* A closing operation essentially consists of a dilation followed by an erosion. In joining, we dilate the contour using a smooth structuring element followed by the application of the

thinning algorithm This operation smooths the contour, fuses narrow breaks and fills gaps on the contour. After joining, we get a BC. It is to be noted that right now we have $\mathcal{R}_{int}(P + \frac{(P-1)}{2})$ BCs Since we are to maintain a fixed population of P BCs, the remaining $\mathcal{R}_{int}(\frac{(P-1)}{2})$ BCs must die. We randomly select P BCs from the present population and let them live, whereas the others die.

3.7 The Mutation

In contrary to traditional GAs, our mutation operation is perhaps the most deterministic step in the entire algorithm. Initially the CBCs are extracted from the P BCs present after the crossover operation. Then, these are introduced in the gradient surface of the gaussian smoothed image Modifications in the boundary of each of the CBCs take place as long as the image energy corresponding to each of them keeps on decreasing A stage comes when the movement of any pixel of a CBC results in an increase in the image energy of the CBC. Then this CBC has got trapped in a local minima and the mutation for this CBC stops. The entire process continues until all the CBCs have got trapped in different local minima of the image energy surface. What follows is a description of the method by which a CBC moves to a local minima by discrete changes in its boundary

The method has a lot of dependence on the location of the pixels of a part of the CBC contained in a 3×3 mask which moves along the CBC. We assign direction codes to all the eight nearest neighbors as shown in Fig 3.12(a). Thus from the point O under consideration, if we move to the top-left pixel, the corresponding direction code is 1; if we move to the exact-top pixel, the direction code is 2 and so on. Now, we note that we can always place a 3×3 mask containing any three successive pixels of a CBC We shall call this the *mutation mask*. We keep on placing the mutation mask over

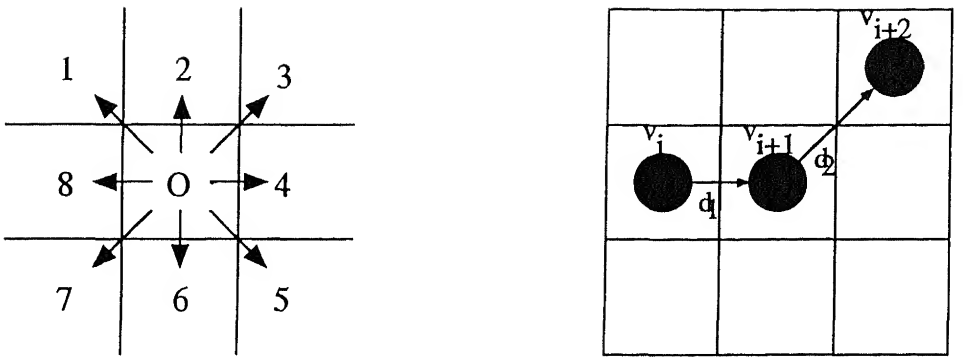


Figure 3.12: (a) The direction code assignment, (b)The mutation mask showing directions d_1 and d_2.

v_i, v_{i+1}, v_{i+2} starting from the root v_1. Let the direction code from v_i to v_{i+1} be d_1, and that from v_{i+1} to v_{i+2} be d_2 (Fig. 3 12(b)). We also define a zonal image energy which forms the parameter for selecting alternative pixel positions inside the mutation mask If G be the set of *on* pixels in the mutation mask, the zonal energy corresponding to the mutation mask is given by

$$E_{zonal} = -\gamma \sum_{v_i \in G} g_i^2 \tag{3.12}$$

Now, we consider the following cases and discuss how pixels are to be changed in order to move the CBC towards the local minima :

I. $d_1 = d_2$: There are two possible subcases under this.

(a) *Both d_1, d_2 are odd* : A typical example of this case is shown in Fig. 3 13 We note that there are two possible alternative paths P_1 and P_2 in addition to the existing path P_0 in between v_i and v_{i+2}. An alternative path is to be considered an *eligible alternative* path provided the contour remains a CBC when it is introduced in place of P_0. Thus, in the figure, P_1 is a possible alternative path whereas P_2 is not. The zonal energy corresponding to the

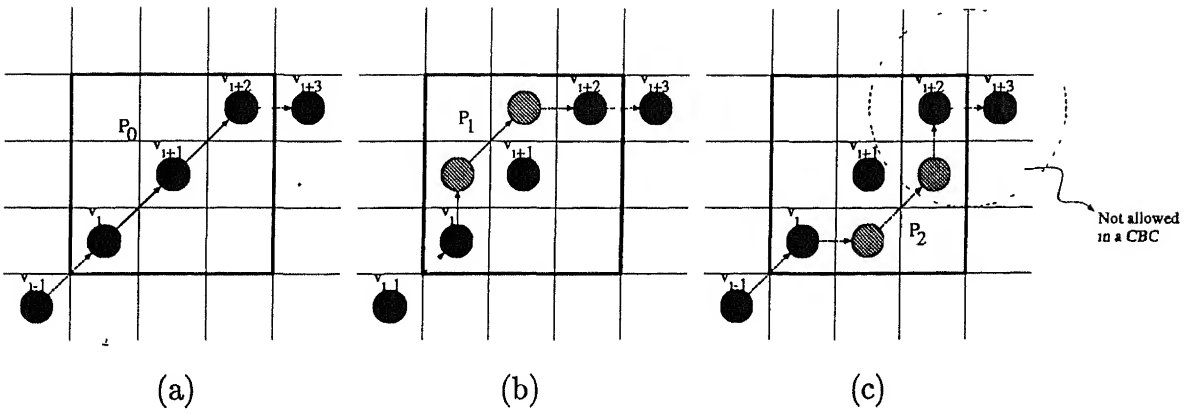


Figure 3.13: $d_1 = d_2$ and both of them are odd. (a) The original path P_0, (b) The eligible alternative P_1, (c) The non-eligible alternative path P_2.

eligible alternative path(s) as well as that for the existing path is computed. The path that minimizes the zonal energy is selected as the path joining v_i and v_{i+1}.

(b) *Both d_1, d_2 are even* : A typical example is shown in Fig 3.14. Once again there are two alternative paths P_1 and P_2 and in this case both are always eligible. So, zonal energy is computed for both of them as well as the original path P_0, and the path with minimum zonal energy is selected.

II. $d_1 \neq d_2$: There are three possible subcases under this.

(a) *d_1 even, d_2 odd* : A typical example of this is shown in Fig. 3.15(a). There is only one possible alternative path P_1, which may or may not be eligible If it is eligible, its corresponding zonal energy is computed along with that of the existing path P_0. The path with a lower zonal energy is selected

(b) *d_1 odd, d_2 even* : A typical example of this case is shown in Fig. 3.15(b). Once again, there is only one possible alternative path P_1, which may or may not be eligible. If it is eligible, its corresponding zonal energy is computed along with that of the existing path P_0. As always, The path with a lower

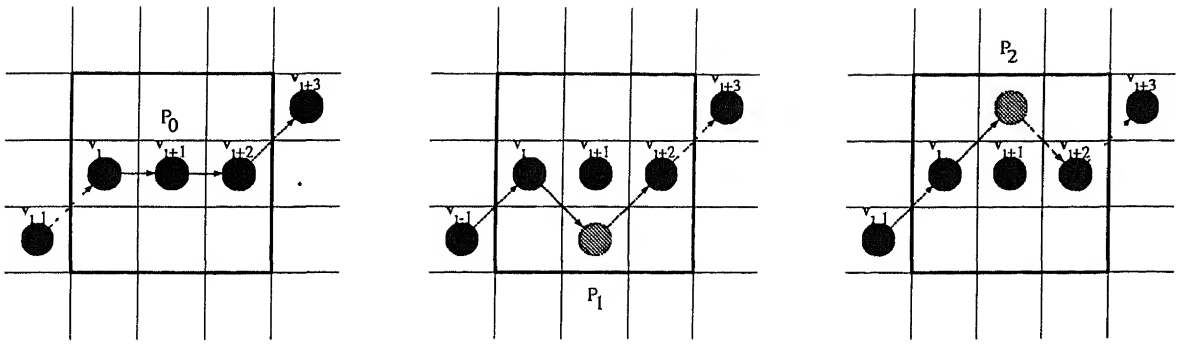


Figure 3.14: $d_1 = d_2$ and both of them are even. (a) The original path P_0, (b) One eligible alternative P_1, (c) Another eligible alternative path P_2.

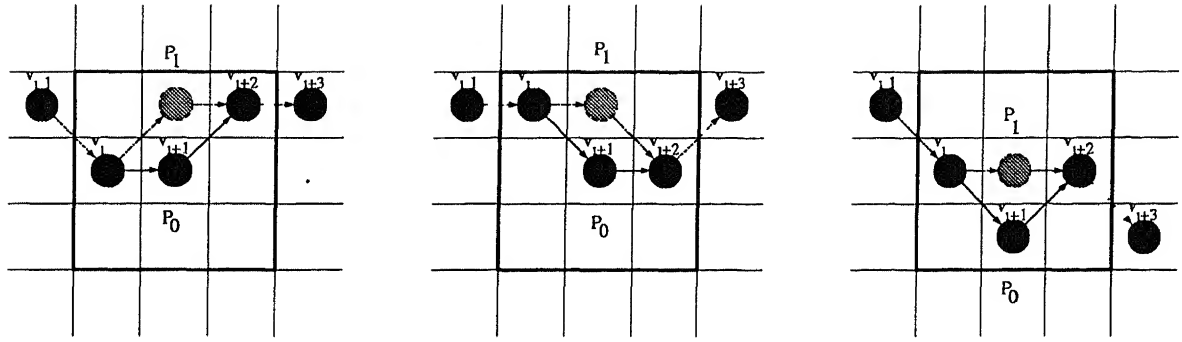
zonal energy is selected.

(c) *d_1 odd, d_2 odd* : A typical example of this is shown in Fig. 3.15(c). There is only one possible alternative path P_1, which is always eligible. So, zonal energy corresponding to this path as well as that of the original path P_0 is computed. The path with a lower zonal energy is selected.

The case that both d_1 and d_2 are even and unequal is not possible in case of a CBC.

This process goes on and the image energy of the CBC keeps on decreasing The CBC is basically going down a valley in the image energy surface. The process comes to a stop when a full cycle along the CBC does not lead to any change in any pixel in the CBC. Now, we have the CBC trapped in a local minima of the image energy landscape.

When the mutation process corresponding to a particular CBC $\mathcal{V}_k$ comes to a stop, the surface of the CBC is very much wrinkled since there was pixel-wise motion during the mutation stage This obviously amounts an increase in the internal energy and hence may lead to an increase in the total snake energy. To prevent this, the wrinkled



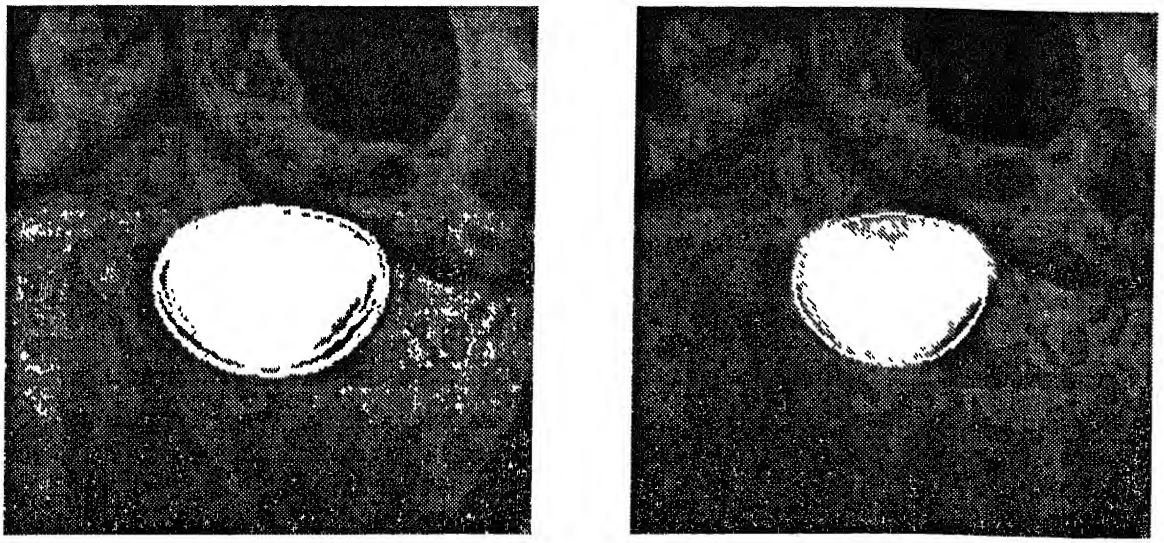
(a) (b) (c)

Figure 3.15· $d_1 \neq d_2$. The original path P_0 and the possible alternative P_1 for (a) d_1 even, d_2 odd, (b) d_1 odd, d_2 even, (c) The original path P_0 and the eligible alternative P_1 for both d_1 and d_2 odd.

CBC is joined, i.e., it is at first dilated followed by a thinning operation. As already mentioned, this smooths out the contour and the process of mutation comes to an end.

3.8 A Discussion

The application of the above steps basically completes one generation Since the initial selection of the CBCs into the mating pool are on the basis of their fitness both in the stochastic remainder selection principle as well as the biased roulette wheel principle, we expect the average energy of the P CBCs in the mating pool to be less than that of the population before the selection process. The basic crossover operation gives birth to a single child keeping both the parents alive. The next step of selecting P CBCs randomly from the mixed population of the parents and the children may actually increase the average energy. Due to its fully random nature, some of the highly fit CBCs may die in this stage causing an increase in the average energy. However, this step is extremely essential since it prevents the population from getting trapped in



(a) (b)

Figure 3 16· (a) A part of the population of the first generation, (b) The best CBC after the first generation

a valley and is very characteristic of a GA. The last stage pushes the CBCs to their nearest energy minima. Thus, the CBCs which lie in the neighborhood of the global minima moves and aligns itself with the position of minimum energy. This CBC may eventually get killed in subsequent generations, but due to a general pattern of decrease in energy the average energy will have a decreasing nature. This will become clear when we look at the plots of the average energy against the generation number. If we keep a track of the minimum energy CBC in each generation we notice that this energy cannot go below a certain value and slowly stabilizes at that particular value. This energy value is identified as the minimum possible value of the energy of a CBC, and the corresponding CBC is identified as the correct boundary of the tumor in the image slice under consideration In the next chapter we shall show the results of applying this algorithm to different image slices along with a discussion and scopes for further improvement of the algorithm.

Chapter 4

Results and Discussion

The basic aim of our boundary detection algorithm is to detect the boundary of the brain tumor in each image slice and separate the tumor in each slice from normal brain tissues Once isolated, the detected tumor in each slice can be further processed for volume measurement and three-dimensional rendering. Now, we shall look at the results of applying the proposed tumor boundary detection algorithm to some MRI data.

4.1 Results

Before applying the actual algorithm, a population of approximate BCs were generated using morphological preprocessing and thinning. The algorithm being computationally involved, the initial population was kept within six to ten BCs. Along with the formation of the initial population, a gradient image of the gaussian smoothed original image was obtained. Some examples of these gradient images are shown in Fig 4.1

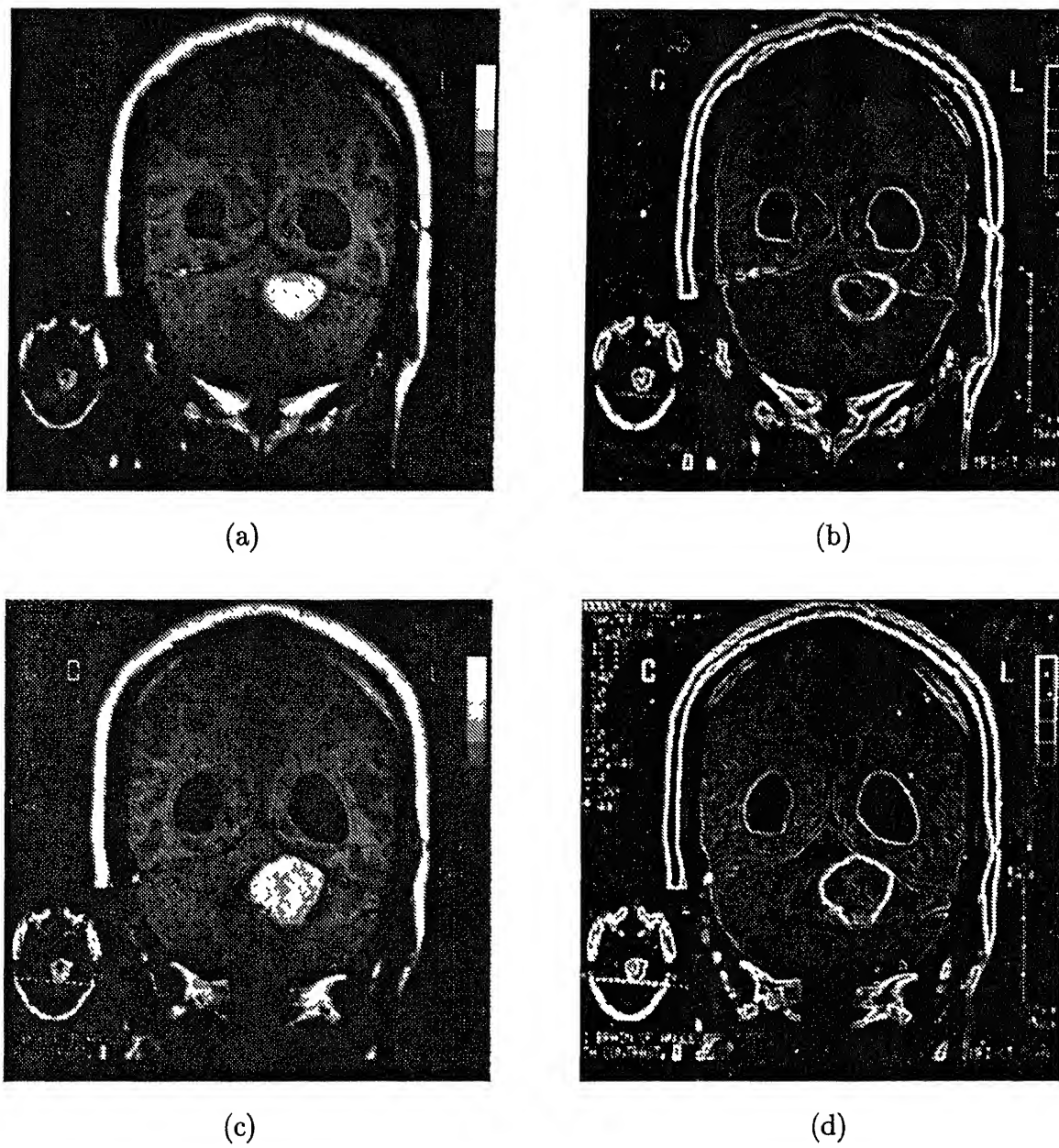


Figure 4.1: Example brain image slices (a),(c) along with their respective gradients (b),(d) after gaussian smoothing

The values of the parameters α, β, γ were determined after some experimentation. The final values of α and β was fixed at 0.1 and that of γ was fixed at 1.0. It is to be noted that we have given more importance to the image energy. Now, we shall take a closer look at the variation of the minimum energy CBC through different generations for the first slice under consideration(Fig. 4.2). Thereafter, we shall show the final detected tumor boundaries, obtained from six slices of the axial MRI data set, superimposed on the gradient image since this will give a more clearer picture of the accuracy of the method We shall also have a look at the variation of the average energy of a population as well as the minimum energy of that population with successive generations. We note that on the whole both the average energy as well as the minimum energy has a decreasing trend. There are occasional increases also which may be attributed mainly to the random selection after the spatial crossover phase. The minimum energy as well as the average energy tend to stabilize to a certain minimum value and we can consider this to be the minimum possible value of the energy function for the given image and the corresponding CBC is identified as the correct boundary of the tumor. There is another interesting thing to noted from the plots. We find that the average energy, at times, especially when the minimum energy has reached the possible minimum for the image, becomes equal to the minimum energy. Now, if it is so that all the CBCs lie in the same position in the energy surface then there is no scope for any decrease of average energy in subsequent generations. This is because of the following reasons - first, the selection operation does not give rise to any CBC in some other position, since it merely selects from the existing CBCs; second, the crossover operator cannot give birth to any CBC outside this position since crossover between two exactly similar contours just gives gives the same contour; third, the mutation operation cannot take any CBC out of its present position since the present position is an energy minima in the local neighborhood. However, we find that often the average energy increases even after becoming equal to the minimum energy. This implies that

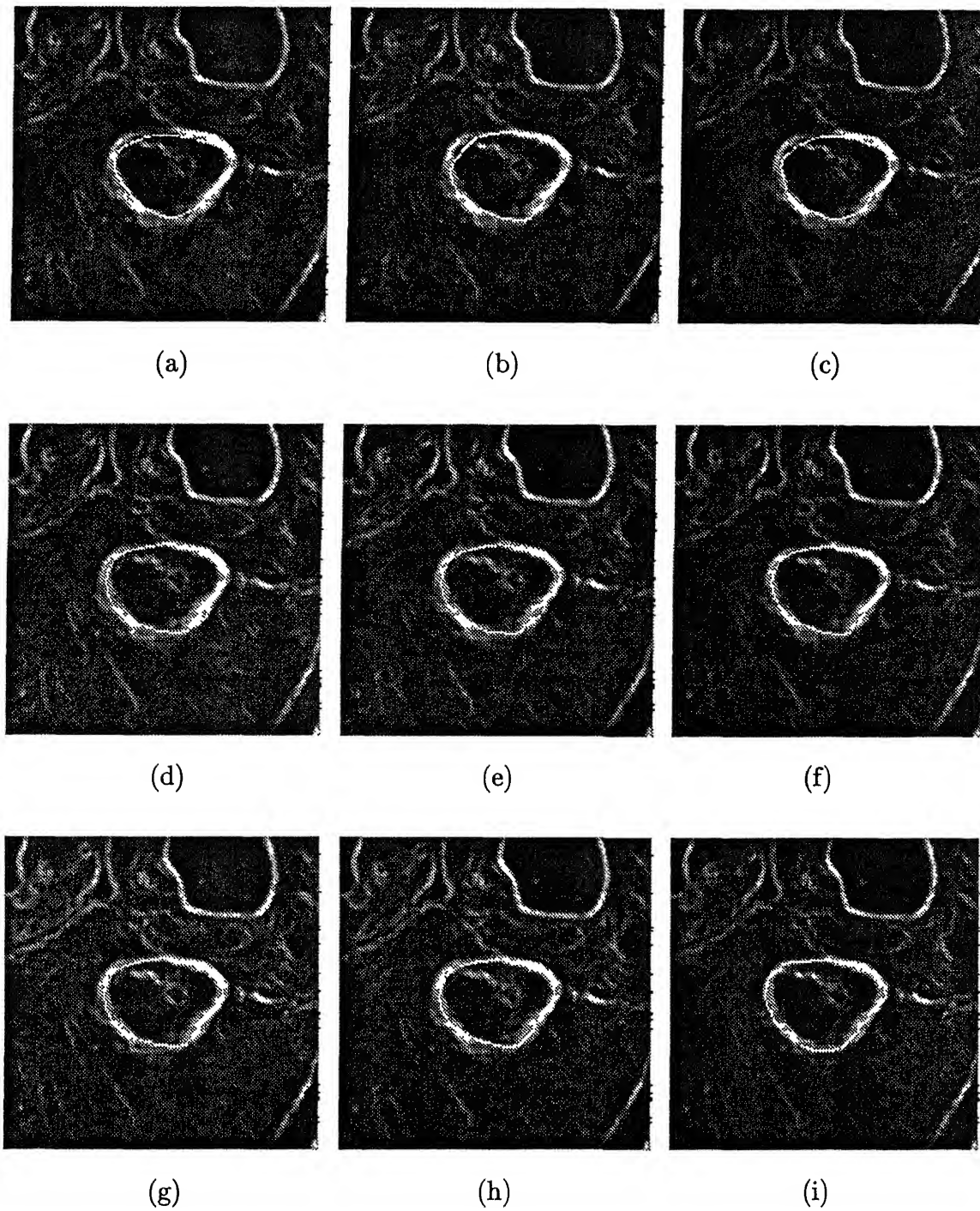
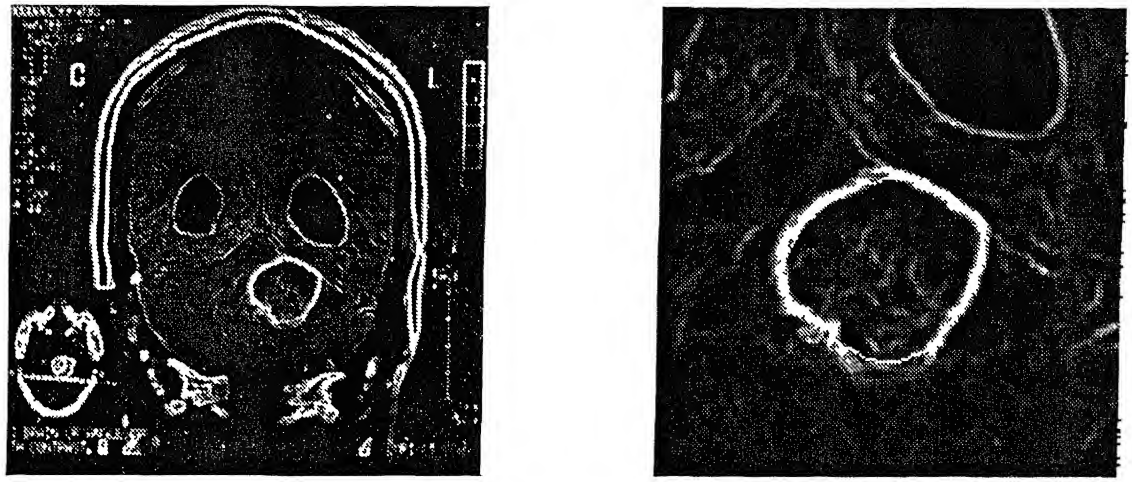


Figure 4 2: The minimum energy CBCs after generation number (a) 1, (b) 3, (c) 4, (d) 8, (e) 9, (f) 15, (g) 19, (h) 44, (i) 49.



(a) (b)

Figure 4 5: (a) The detected tumor boundary superimposed on the gradient image, (b) The tumor region in (a) zoomed to get a clear view

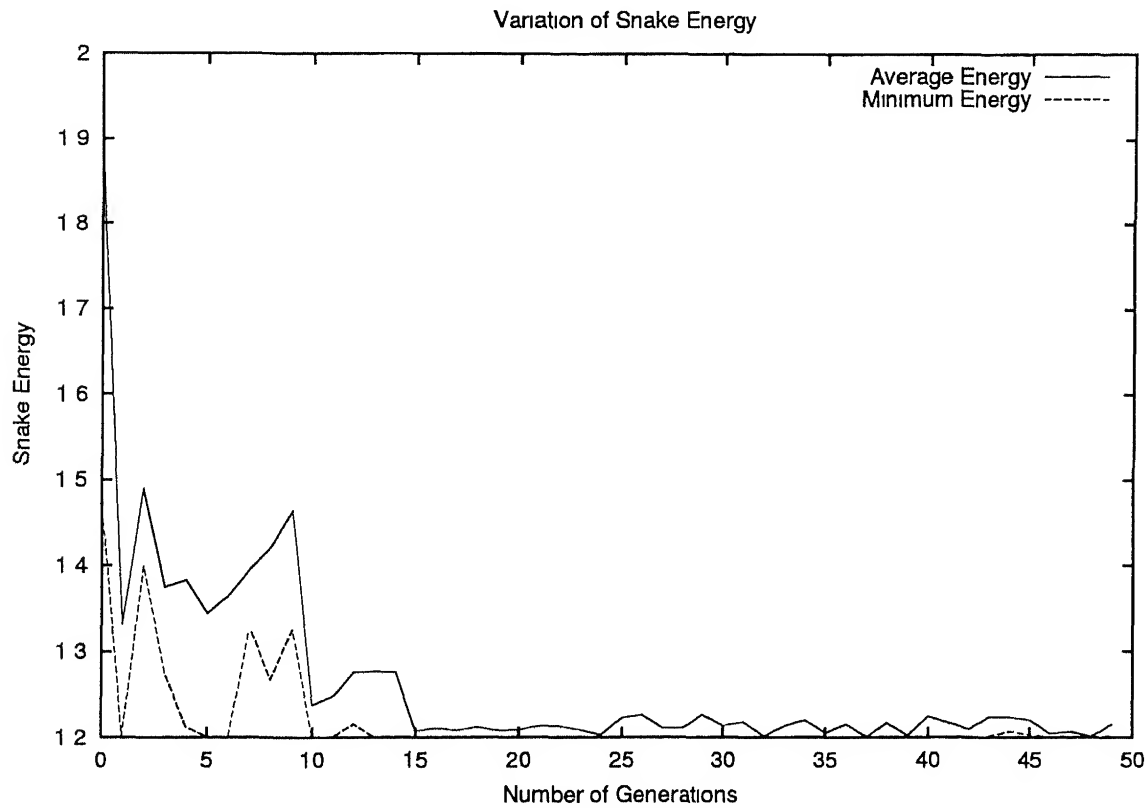
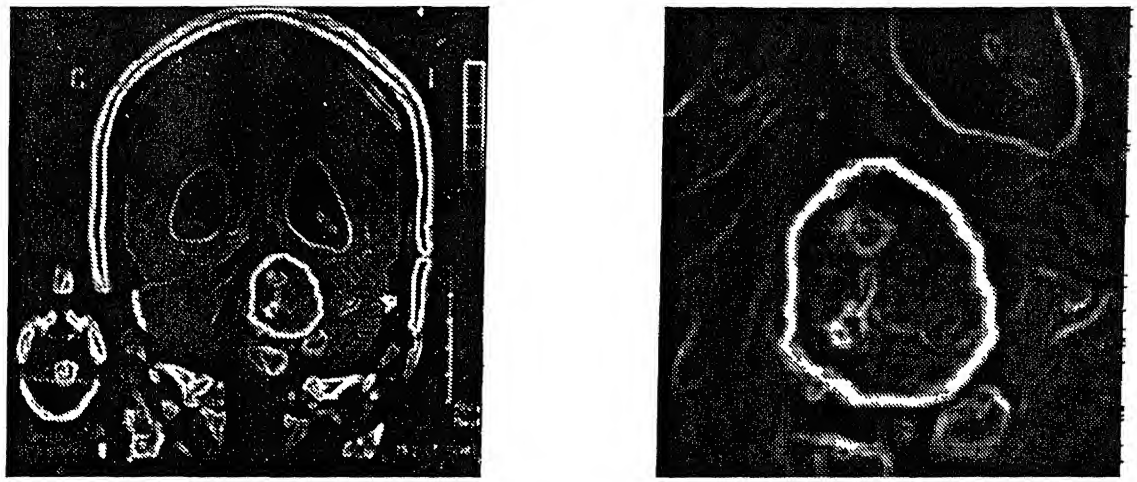


Figure 4.6: The variation of the average and minimum energy of the population.



(a) (b)

Figure 4.7: (a) The detected tumor boundary superimposed on the gradient image, (b) The tumor region in (a) zoomed to get a clear view

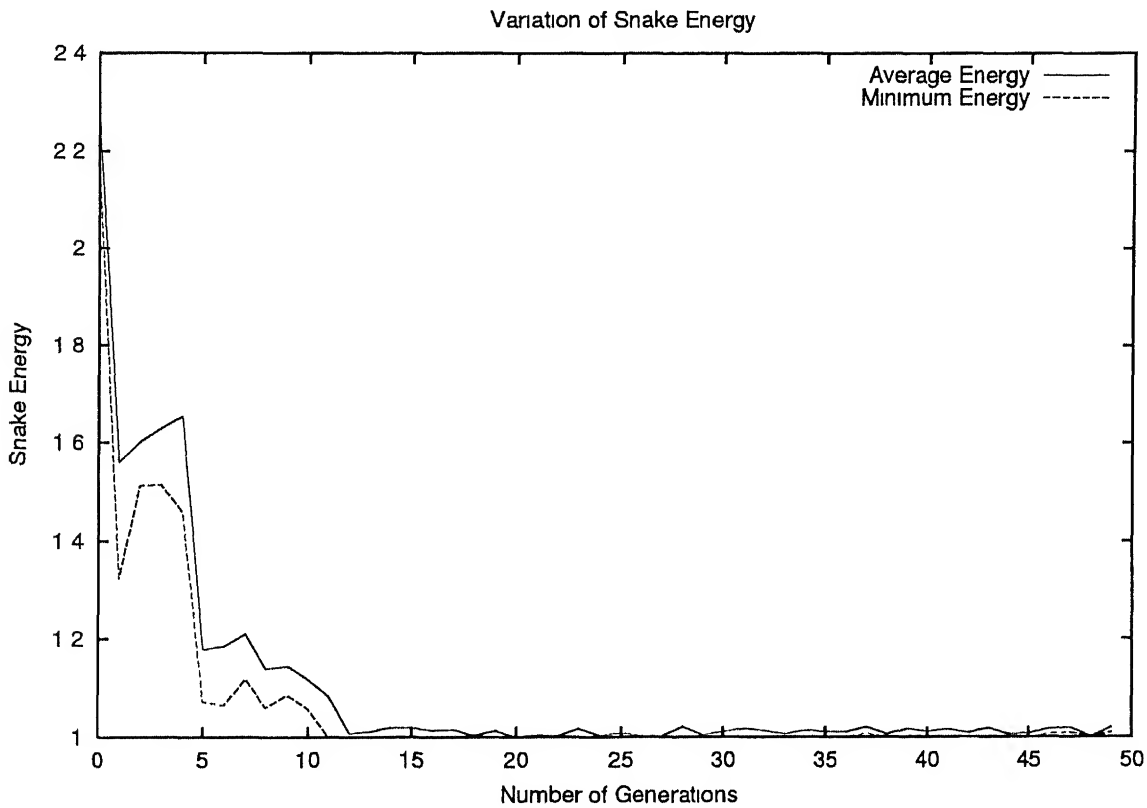


Figure 4.8: The variation of the average and minimum energy of the population.

(a) (b)

Figure 4.9: (a) The detected tumor boundary superimposed on the gradient image, (b) The tumor region in (a) zoomed to get a clear view.

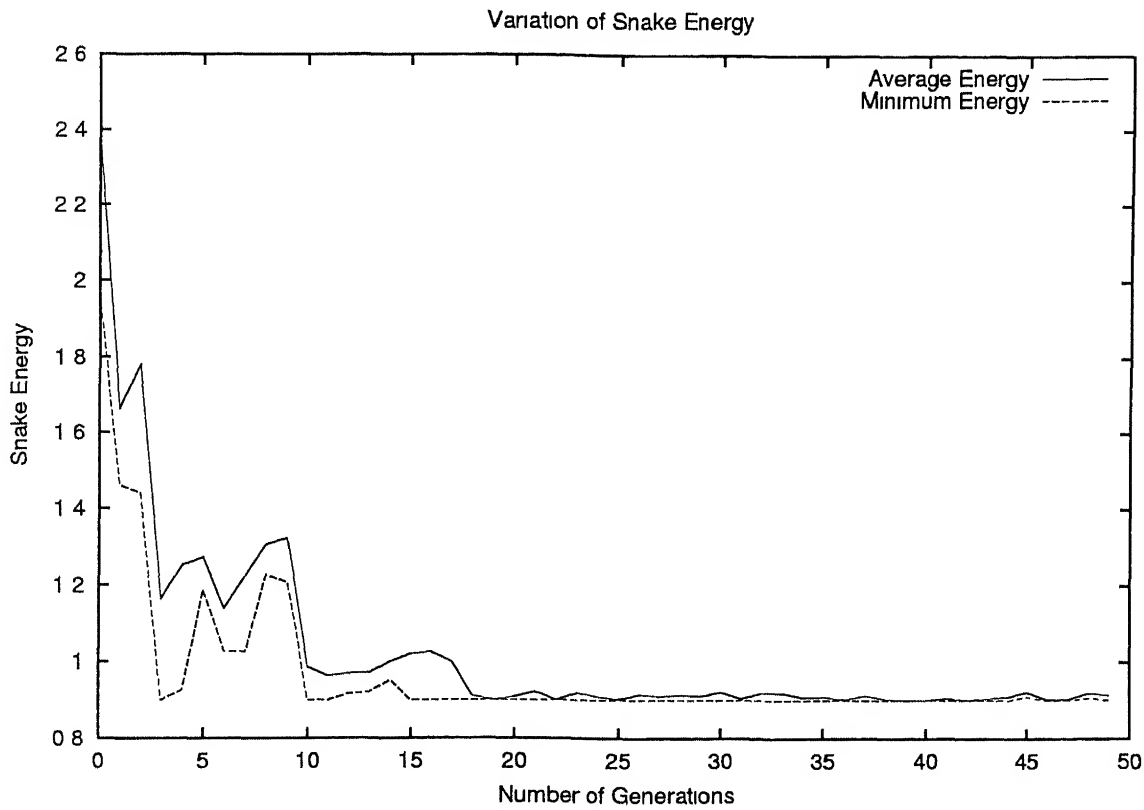


Figure 4.10: The variation of the average and minimum energy of the population.

(a) (b)

Figure 4.11: (a) The detected tumor boundary superimposed on the gradient image, (b) The tumor region in (a) zoomed to get a clear view.

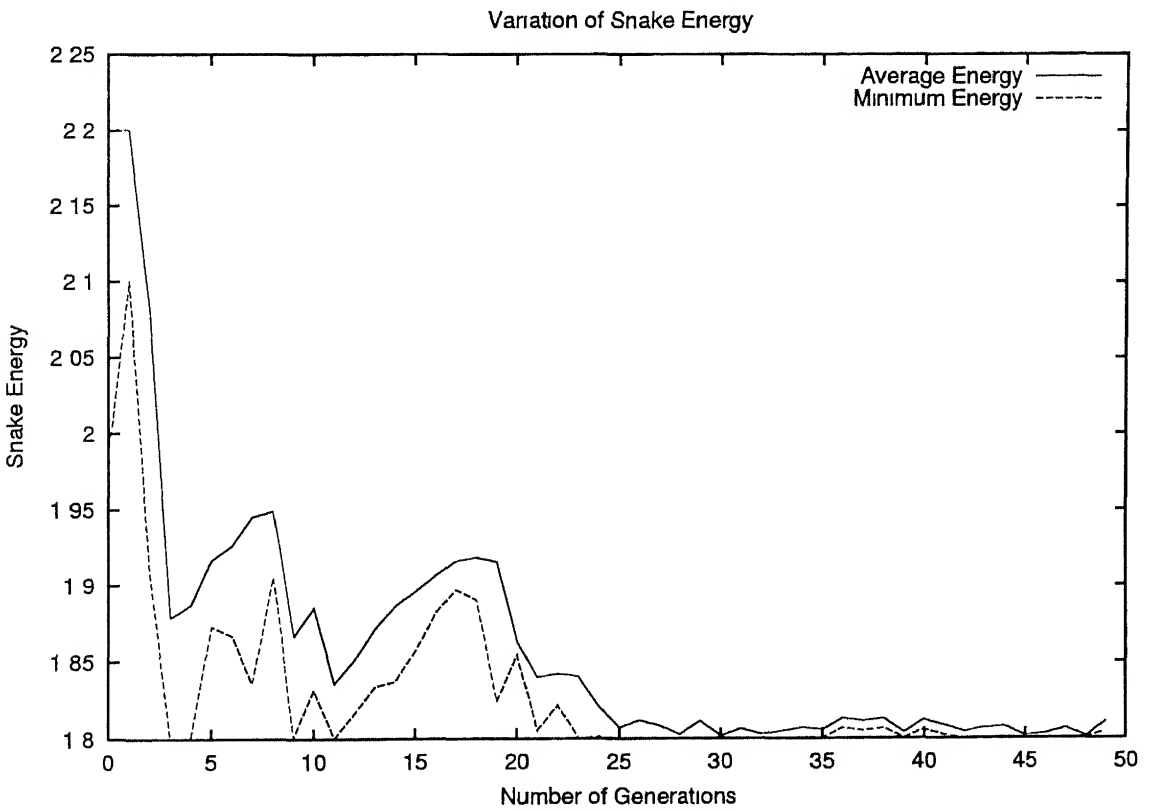
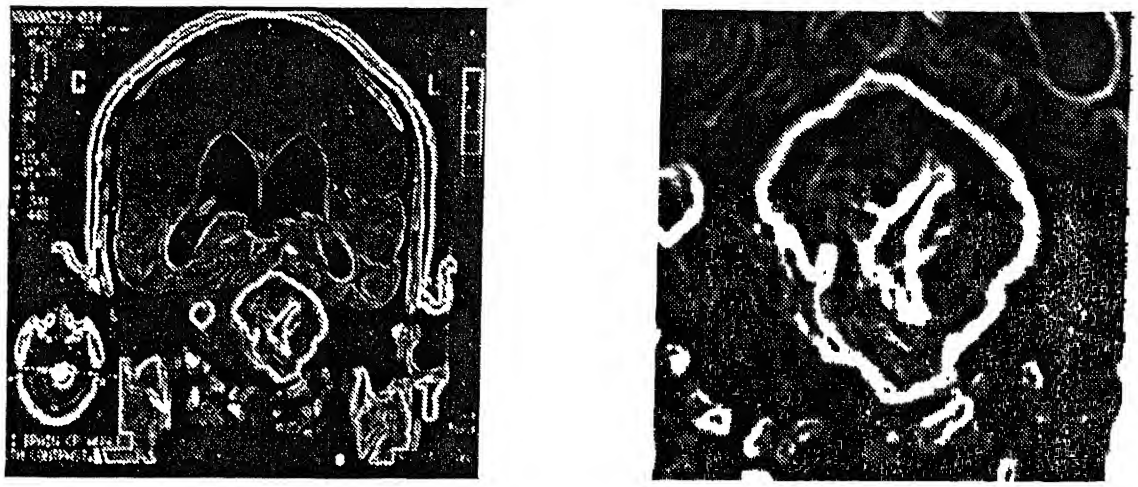


Figure 4.12: The variation of the average and minimum energy of the population



(a) (b)

Figure 4 13: (a) The detected tumor boundary superimposed on the gradient image, (b) The tumor region in (a) zoomed to get a clear view.

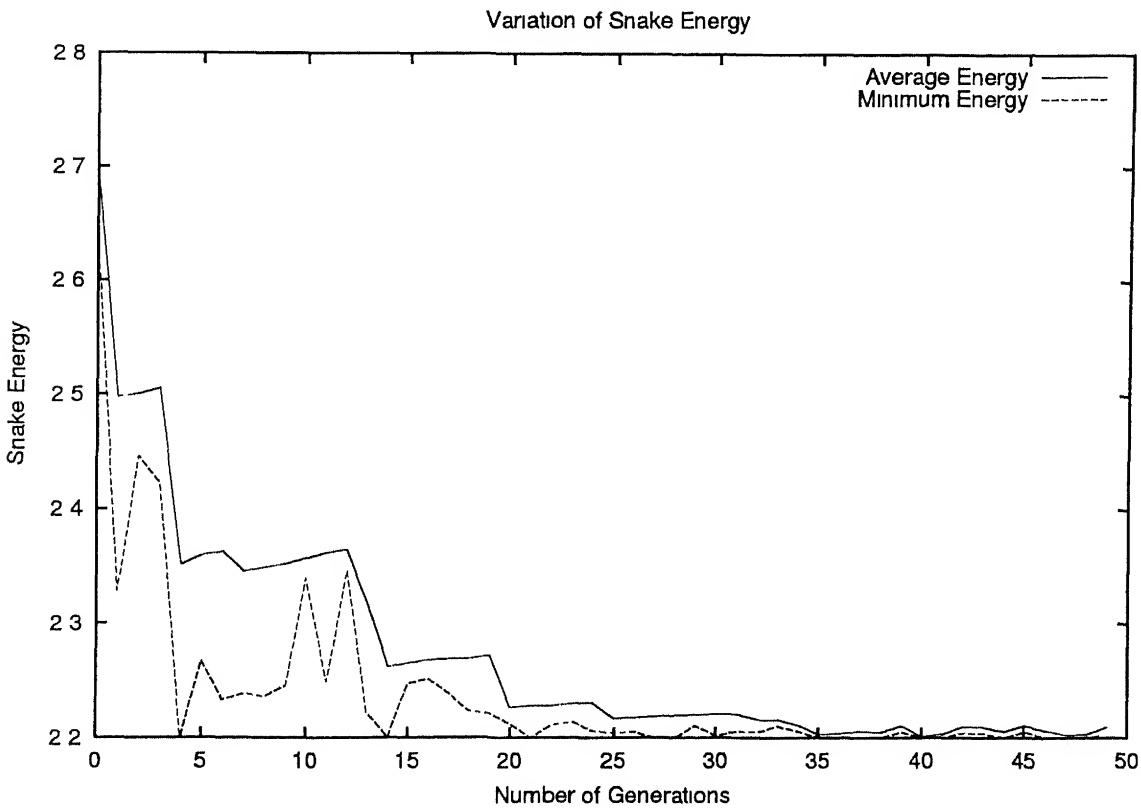


Figure 4 14: The variation of the average and minimum energy of the population

all the CBCs although having the same energy do not lie in the same position and hence the energy surface must have more than one position which gives the desired minimum energy This is not difficult to explain if we take a closer look at the gradient images. In the region surrounding the actual tumor boundary, the gray level of the image is almost the same and thus all closely located CBCs in the neighborhood will have equal or almost equal values for the image energy. Also, since the 'amount of bending' is almost the same, they will have almost identical values of the internal energy

4.2 Discussion

The work presented in this thesis is perhaps the first effort of applying a variant of the genetic algorithm to medical image segmentation problems. Preliminary evaluation of the performance of this method in automatic brain tumor boundary detection shows that it is a promising approach.

- The approach is extremely robust. Since the algorithm deals with a population of possible solutions in the form of CBCs, there is very little chance that the entire population gets stuck in a local minima. Even if a few CBCs get trapped in a local minima the algorithm does not stop. In fact the CBCs lying outside the local minima pull out these CBCs from the valley or in other words, give rise to children who do not fall in the local minima.

- The suitability of the final solution is tested over and over. Even if a CBC has reached a deep valley, which may actually be a global minima, it may get killed after the crossover stage Then the whole population may have to evolve through generations to get to the solution once again. If the same solution is reached each time, then that is considered as the desired contour. However, if we get

a better solution in future generations, we will understand that the solution we had previously got was just a local minima.

- There is no dependence on the selection of initial population. As long as some of the CBCs are in the broad neighborhood of the actual tumor, which the preprocessing stage, however approximately it is done, ensures, the correct solution will evolve after some generations.

- The method is extremely insensitive to noise because of the morphological preprocessing. Thresholding followed by erosion removes any noise and other brain tissue structures in which we are not interested. The dilation operation restores a part of the tumor which had been removed during erosion.

- The gaussian smoothing of the image before computing the image gradient prevents the formation of unwanted valleys due to the presence of noise in the original image.

- Since the algorithm deals with a number of possible solutions and does not stop after getting a local minima, it is computationally expensive.

4.3 Scope for future work

As already mentioned, this is perhaps the first attempt of its type to detect brain tumor boundary using GAs. Several steps in the algorithm need to be further investigated and refined. The morphological preprocessing stage in the present form of the algorithm is semi-automatic. The structuring elements in the morphological operations should be selected automatically and adaptively. Several design parameters are there which need to be properly estimated The population size P is a very important parameter and needs further investigation. The question of whether to keep a fixed population or a

[7] J. G. Thomas, R. A. Peters II and P. Jeanty, "Automatic Segmentation of Ultrasound Images Using Morphological Operators", *IEEE Trans. Med. Imag.*, vol. 10, no. 2, pp. 18-186, 1991.

[8] C. T. Chen, E. Tsao, and W. C. Lin, "Medical image segmentation by a constraint satisfaction neural network", *IEEE Trans. Nucl. Sci.*, vol. 38, no. 2, pp. 678-686, 1991.

[9] A. Yezzi, Jr., S. Kichenassamy, A. Kumar, P. Oliver and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery", *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 199-209, April, 1997.

[10] R. Malladi, J. A. Sethian, B. C. Vemuri, "Shape Modelling with Front Propagation: A Level Set Approach", *IEEE Trans. Pat. Anal. Mac. Intell.*, vol. 17, no. 2, pp. 158-174, February, 1995.

[11] R. M. Haralick, S. R. Sternberg and X. Zhuang, "Image analysis using mathematical morphology", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 4, pp. 532-550, July, 1987.

[12] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall of India, 1995.

[13] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, 1987.

[14] Edited by S. K. Pal and P. P. Wang, *Genetic Algorithms for Pattern Recognition*, CRC Press, 1996.

[15] K. Deb, *Optimization for Engineering Design: Algorithms and Examples* Prentice Hall of India, 1995.

[16] P. Baxandall and H. Lieback, *Vector Algebra*, Oxford University Press, 1986.